

Työajanseuranta-mobiilisovellus

Case: Aamos Group Oy

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2016
Toni Heikkilä

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

HEIKKILÄ, TONI:

Työajanseuranta-mobiilisovellus
Case: Aamos Group Oy

Ohjelmistotekniikan opinnäytetyö, 55 sivua

Kevät 2016

TIIVISTELMÄ

Työn tavoitteena oli toteuttaa mobiilisovellus helpottamaan Aamos Group Oy:n kehittämän kalenteripohjaisen työnajanseuranta-palvelun käyttöä älypuhelimella. Työajanseuranta-sovelluksesta kehitettiin natiivit mobiilisovellukset Android- ja Windows Phone -mobiilikäyttöjärjestelmille ja ne julkaistiin sovelluskaupoissa.

Aamos Group Oy:n työajanseuranta-palvelun tarkoituksena on tehdä erilaisten työmerkintöjen, kuten työtuntien, kilometrien ja päivärahojen, kirjaamisesta helpompaa työntekijälle, kun työmerkinnät kirjataan samaan kalenteriin työtehtävien kanssa. Palvelu helpottaa myös työnantajaa työaikalain velvoittaman työaikakirjanpidon ylläpitämisessä luomalla kullekin työntekijälle omat seurantaraportit. Palvelusta saatavia seurantaraportteja voidaan käyttää myös palkanmaksun perusteena.

Työajanseuranta-mobiilisovellukseen luotiin työmerkintöjen kirjaamista helpottavia toimintoja. Sovelluksen avulla on esimerkiksi mahdollista kirjata kaikki tarvittavat merkinnät yhden näkymän kautta, kun normaalisti ne olisi kirjattava yksitellen. Lisäksi työtehtäviä luotaessa kalenteriin käyttäjällä on mahdollisuus lisätä tehtävän kesto automaattisesti päivän työtunteihin. Muun muassa näiden ominaisuuksien avulla palvelun käyttökokemusta pyrittiin parantamaan älypuhelimella.

Mobiilisovelluksen toteuttamisen lisäksi työssä tutustuttiin Android- ja Windows Phone -mobiilisovelluskehitykseen sekä kyseisten mobiilikäyttöjärjestelmien tärkeimpiin ominaisuuksiin mobiilisovelluskehityksen kannalta. Työajanseuranta-mobiilisovelluksen osalta keskeisintä oli tutkia kalenterimerkintöjen hallintaa mobiilikäyttöjärjestelmien tarjoamien rajapintojen avulla. Mobiilisovelluksen julkaisemiseksi sovelluskauppoihin oli tutustuttava siihen liittyviin vaatimuksiin ja käytäntöihin.

Asiasanat: työajanseuranta, Android, Windows Phone, mobiilisovellus

Lahti University of Applied Sciences
Degree Programme in Information Technology

HEIKKILÄ, TONI:

Mobile app for tracking working time
Case: Aamos Group Oy

Bachelor's Thesis in Software Engineering, 55 pages

Spring 2016

ABSTRACT

The objective of the thesis was to make a mobile application to ease the use of the calendar-based working time tracking service of Aamos Group Oy on smartphones. The native mobile applications of the service were developed for Android and Windows Phone mobile operating systems and they were released in the application stores.

The purpose of the working time tracking service developed by Aamos Group Oy is to make it easier for employees to write down different kinds of entries such as working hours, mileage and daily allowances. The entries are written down in the same calendar as work assignments. The employer can use the service to create a working hours register for each employee as stipulated in the working hours act. The working hours register can also be used as the basis of the salary.

Many features were created to the working time tracking mobile application to make it easier to write down entries. Users can for example write down all entries on the same screen with the mobile application when normally they should be made separately. Additionally, the user can add the duration of an event automatically to the working hours of the day while creating a new work assignment. These and other functions were developed to improve the user experience of the service on smartphones.

In addition to developing the mobile application, Android and Windows Phone mobile application development and the most important features of them were explored in the thesis. The most crucial aspect while developing the mobile application was to investigate how to manage calendar events using the interfaces of the mobile operating systems. The requirements and policies of the application stores had to be studied before the mobile applications could be released in the stores.

Key words: working time tracking, Android, Windows Phone, mobile app

SISÄLLYS

1	JOHDANTO	1
2	MOBIILISOVELLUSKEHITYS	2
2.1	Android	2
2.1.1	Kehitysympäristö	2
2.1.2	Aktiviteetit	3
2.1.3	Aktiviteetin elinkaari	4
2.1.4	Sovellusvaatimustiedosto	5
2.2	Windows Phone	8
2.2.1	Kehitysympäristö	8
2.2.2	Sivut	9
2.2.3	Sovelluksen elinkaari	10
2.2.4	Sovellusvaatimustiedosto	11
3	KALENTERIRAJAPINTOJEN KÄYTTÖ	14
3.1	Android	14
3.1.1	Kalenterimerkintöjen hakeminen	14
3.1.2	Kalenterimerkinnän luominen	16
3.1.3	Kalenterimerkinnän päivittäminen	17
3.1.4	Kalenterimerkinnän poistaminen	18
3.1.5	Kalenteritietojen hallinta kalenterisovelluksen avulla	18
3.1.6	Kalenterien hakeminen	20
3.2	Windows Phone	21
3.2.1	Kalenterimerkintöjen hakeminen	21
3.2.2	Kalenterimerkinnän luominen	22
3.2.3	Kalenterimerkinnän päivittäminen	23
3.2.4	Kalenterimerkinnän poistaminen	23
3.2.5	Kalenterien hakeminen	24
4	JULKAISEMINEN SOVELLUSKAUPPAAN	25
4.1	Google Play -kauppa	25
4.1.1	Kehittäjätilin luonti	25
4.1.2	Sovelluksen asennustiedoston luonti	26
4.1.3	Sovelluksen julkaiseminen kauppaan	28
4.2	Windows Phone -kauppa	33
4.2.1	Kehittäjätilin luonti	33

4.2.2	Sovelluksen asennustiedoston luonti	34
4.2.3	Sovelluksen julkaiseminen kauppaan	37
5	SOVELLUKSEN TOTEUTUS	43
6	YHTEENVETO	49
	LÄHTEET	50

1 JOHDANTO

Työnantaja on työaikalain mukaan velvoitettu kirjaamaan työntekijöiden tekemät työtunnit ja niistä suoritettut korvaukset. Työtuntien lisäksi kirjanpidosta tulee käydä ilmi työntekijän lisä-, yli-, hätä- ja sunnuntaityötunnit. Työnantajan on säilytettävä työntekijän työaikakirjanpito seuraavan kahden vuoden ajan. Työaikakirjanpidon laiminlyönti johtaa sakkoon työaika-rikkomuksesta. (Työaikalaki 605/1996, 37 §, 38 §, 42 §.)

Myrskyläläinen työvoima- ja ICT-palveluihin erikoistunut Aamos Group Oy on kehittänyt kalenteripohjaisen työajanseurannan työaikakirjanpidon ylläpitämistä varten. Työntekijä kirjaa tekemänsä työtunnit Microsoftin Office 365 -kalenteriin niille valmiiksi asetettuihin kalenterimerkintöihin. Työtuntien lisäksi kalenteriin voidaan kirjata esimerkiksi erilaiset lisät, kuten iltaja sunnuntailisät sekä kilometrit ja päivärahat. Samaan kalenteriin työntekijä voi kirjata myös sekä tehdyt että tekemättömät työtehtävänsä, mikä helpottaa työntekijän työajankäytön suunnittelua. Palvelusta on mahdollista saada Excel-taulukkolaskentamuotoiset työajanseurantaraportit, jotka helpottavat tehdyn työn seuraamista ja palkanmaksamista.

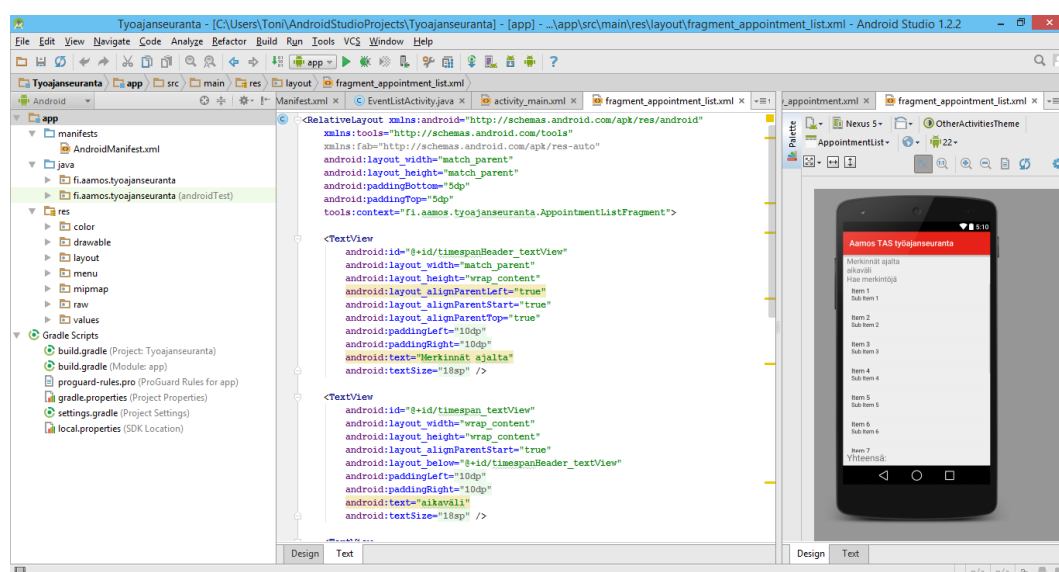
Työn tarkoituksena on luoda Android- ja Windows Phone -mobiilikäyttöjärjestelmille natiivi mobiilisovellus, joka tekee työaikakirjanpidon ylläpitämisestä ja seuraamisesta vaivattomampaa älypuhelimella. Sovellukseen on tarkoitus luoda käyttöä helpottavia ominaisuuksia, kuten kaikkien kirjattavien tietojen kirjaaminen kalenteriin yhdellä lomakkeella, uuden työtehtävän lisäämisen yhteydessä sen keston lisääminen automaattisesti päivän työtunteihin ja paikkatiedon lisääminen kalenterimerkintään. Työssä perehdytään Android- ja Windows Phone -mobiilikäyttöjärjestelmien mobiilisovelluskehitykseen sekä tutustutaan kalenterirajapintojen käyttöön ja mobiilisovelluksien julkaisemiseen sovelluskauppoihin.

2 MOBIILISOVELLUSKEHITYS

2.1 Android

2.1.1 Kehitysympäristö

Android-mobiilikäyttöjärjestelmälle sovellusten kehittäminen tapahtuu Googlen kehittämällä ja JetBrainsin IntelliJ IDEA -ohjelmistoon pohjautuvalla Android Studio -sovelluskehitysympäristöllä, joka esiintyy kuviossa 1. Android Studio tarjoaa työkalut sekä itse ohjelmakoodin että sovelluksen ulkoasun luomiseen. Tämän lisäksi sen avulla voidaan muun muassa testata sovelluksia virtuaalilaitteissa emulaattorin avulla tai fyysisessä, tietokoneeseen yhdistetyssä laitteessa sekä luoda asennuspaketti sovelluksen asennusta ja Google Play -kauppaan julkaisemista varten. Android Studio -kehitysympäristö on käytettävissä maksutta. (Android Developers 2016b.)



KUVIO 1. Android Studio -sovelluskehitysympäristö

Android-sovellusten ohjelmoinnissa käytetään Java-ohjelmointikieltä. Sovelluksen käyttöliittymän ulkoasu luodaan XML-merkintäkieltä käyttäen. (Android Developers 2016h.)

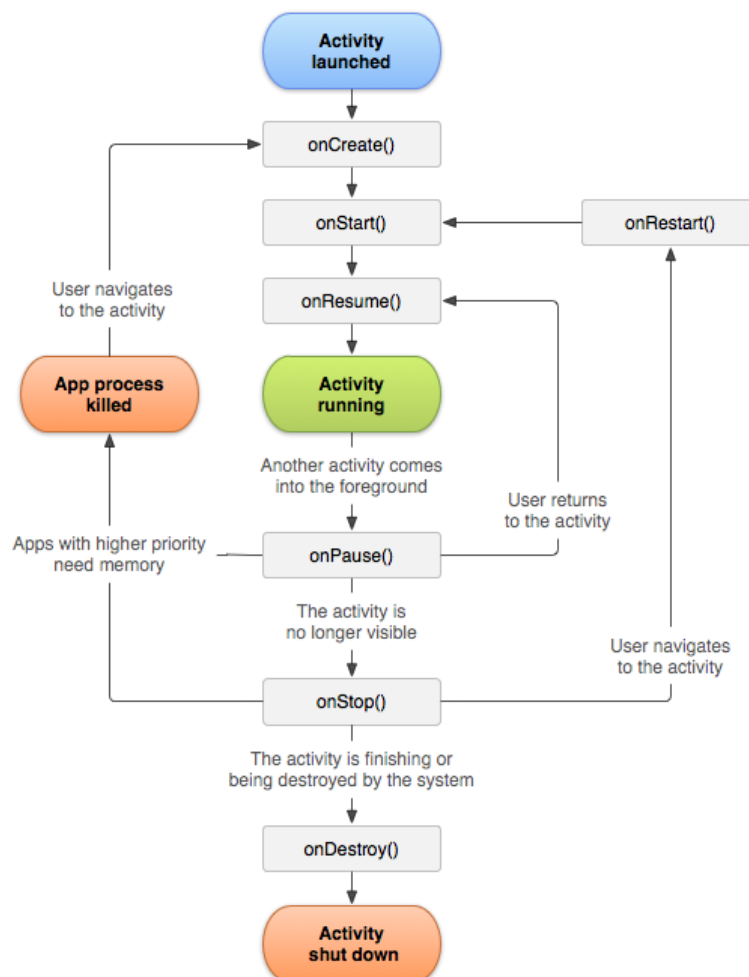
2.1.2 Aktiviteetit

Android-kehityksessä sovelluksen käyttöliittymän näkymiä kutsutaan aktiviteeteiksi. Näiden avulla käyttäjät ovat vuorovaikutuksessa sovelluksen ja laitteiston toimintoihin. Sovellukset koostuvat useasta eri aktiviteetista, mutta sovelluksella on kuitenkin tyypillisesti pääaktiviteetti, joka on sovelluksen aloitusnäky. Pääaktiviteetin määrittäminen tapahtuu sovellusvaatimustiedostossa. Aktiviteetit voivat lisäksi käynnistää uusia aktiviteetteja, joissa suoritetaan erilaisia toimintoja. Käynnistettävät aktiviteetit voivat olla myös jonkin toisen sovelluksen aktiviteetteja, jolloin kaikkia sovelluksen toiminnallisuuksia ei tarvitse luoda itse, vaan voidaan hyödyntää jo olemassa olevia sovelluksia. Tätä voitaisiin hyödyntää esimerkiksi kamera-sovelluksessa, joka käynnistäisi sähköpostisovelluksen viestin lähetyksen aktiviteetin juuri otetun kuvan jakamista varten. Luodut aktiviteetit perivät Activity-luokan ja sen ominaisuudet, kuten onCreate- ja onPause-metodit, joita käytetään aktiviteetin luomiseen ja keskeyttämiseen sekä niiden vaatimien toiminnallisuuksien toteuttamiseen. (Android Developers 2016a.)

Aktiviteettien ulkoasu luodaan erilaisten käyttöliittymäelementtien avulla XML-merkintäkielellä. Yleisiä elementtejä ovat esimerkiksi tekstikentät, erilaiset painikkeet, listat sekä pudotusvalikot. Jokaisella elementillä on lisäksi omia määreitä, joita voivat olla esimerkiksi elementin yksilöivä tunnistus, teksti, korkeus sekä leveys, väri ja marginaali. Ulkoasun luominen on mahdollista myös ohjelmallisesti, ja sitä käytetään yleensä tapauksissa, joissa jonkin toiminnon halutaan muuttavan käyttöliittymässä esiintyvien elementtien määreitä. Elementtien asetteluun käytetään erilaisia sijoittelutyppejä, joista yleisimmät ovat lineaarinen ja suhteellinen sijoittelu. Lineaarisessa sijoittelussa elementit asetellaan riviin joko horisontaalisesti tai vertikaalisesti. Suhteellisessa sijoittelussa sen sijaan elementit nimensä mukaisesti sijoitellaan suhteessa toisiinsa, jolloin elementeille määritetään sijainti jonkin toisen elementin vieressä. (Android Developers 2016j.)

2.1.3 Aktiviteetin elinkaari

Androidin aktiviteetit kerääntyvät pinoon, jonka päällimmäisenä on aktiviteetti, joka on aktiivinen ja näkyvissä käyttäjälle. Aktiviteetilla on kolme perustilaa. Edellä mainitun aktiivisena olemisen lisäksi aktiviteetti voi olla keskeytetty tai pysäytetty. Aktiviteetti on keskeytetty silloin, kun se on vielä näkyvissä, mutta toinen aktiviteetti on kuitenkin aktiivisena. Aktiviteetti pysäytetään tapauksissa, joissa toinen aktiviteetti on yksinään näkyvissä käyttäjälle. Keskeytetyt ja pysäytetyt aktiviteetit ovat yhä muistissa, mutta järjestelmä voi kuitenkin tuhota ne, mikäli muisti on vähissä ja sitä on vapautettava muihin toimintoihin. Pinon päällimmäisenä oleva aktiviteetti tuhoetaan ja poistetaan pinosta käyttäjän painaessa takaisin-painiketta, milloin edellinen aktiviteetti palautuu aktiiviseksi. (Android Developers 2016a.)



KUVIO 2. Aktiviteetin elinkaari (Android Developers 2016a)

Aktiviteetin luominen ja käynnistäminen aloitetaan kuvion 2 mukaisesti kutsumalla Activity-luokasta perittyä onCreate-metodia. Tässä metodissa on tarkoitus alustaa kaikki aktiviteetissa käytettävät komponentit ja asettaa aktiviteetille sille luotu ulkoasu käyttöliittymää varten. Aktiviteetin varaamat resurssit on hyvä vapauttaa onDestroy-metodissa, joka suoritetaan aktiviteettiä tuhohtaessa. Koko aktiviteetin elinkaari tapahtuu näiden kahden edellä mainitun metodin välissä. (Android Developers 2016a.)

Aktiviteetti on näkyvissä käyttäjälle onStart- ja onStop-metodien välisenä aikana. Metodia onStart kutsutaan hetkellä, jolloin aktiviteetti on tulossa näkyviin, ja onStop-metodia silloin, kun aktiviteetti ei ole enää näkyvissä käyttäjälle. Tänä aikana hallitaan resursseja, joita käyttäjälle näytetään aktiviteetin välityksellä. Metodia onRestart kutsutaan aktiviteetin tullessa uudelleen näkyviin sen pysäyttämisen jälkeen. Aika, jolloin aktiviteetti on käyttäjän käytettävissä, tapahtuu metodien onResume ja onPause välillä. Mikäli järjestelmä on joutunut tuhoamaan aktiviteetin vapauttaakseen muistia, kutsutaan aktiviteetin luovaa onCreate-metodia uudelleen aktiviteetin tullessa jälleen aktiiviseksi. Aktiviteetti voidaan tuhota ainoastaan onPause-, onStop- ja onDestroy-metodien jälkeen. (Android Developers 2016a.)

2.1.4 Sovellusvaatimustiedosto

Jokaisessa Android-sovelluksessa on oltava kuvion 3 mukainen sovellusvaatimustiedosto. Sen avulla sovelluksesta esitetään tärkeitä tietoja ja vaatimuksia sekä Android-järjestelmälle että laitteelle. Tiedosto on nimettävä AndroidManifest-tiedostoksi, ja sen tiedostomuoto on oltava XML. (Android Developers 2016c.)

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="fi.aamos.tyoajanseuranta" >
4
5      <uses-permission android:name="android.permission.READ_CALENDAR" />
6      <uses-permission android:name="android.permission.WRITE_CALENDAR" />
7      <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
8      <uses-permission android:name="android.permission.INTERNET" />
9
10     <application
11         android:allowBackup="true"
12         android:icon="@mipmap/ic_launcher"
13         android:label="@string/app_name"
14         android:theme="@style/OtherActivitiesTheme" >
15         <activity
16             android:name=".MainActivity"
17             android:configChanges="orientation|keyboardHidden|screenSize"
18             android:label="@string/app_name"
19             android:theme="@style/MainActivityTheme" >
20             <intent-filter>
21                 <action android:name="android.intent.action.MAIN" />
22
23                 <category android:name="android.intent.category.LAUNCHER" />
24             </intent-filter>
25         </activity>
26         <activity...>
27         <activity...>
28         <service...>
29     </application>
30
31 </manifest>

```

KUVIO 3. Android-sovelluksen vaatimustiedosto

Sovellusvaatimustiedostossa nimetään kehitettävä sovelluspaketti kuvion 3 rivin 3 osoittamalla tavalla. Tämän avulla sovellukselle saadaan yksilöllinen tunniste. Lisäksi sovellukselle määritetään sovellusvalikossa näkyvä nimi ja kuvake sekä muita sovelluksen mahdollisesti tarvitsemia tietoja. Komponentit, kuten aktiviteetit ja palvelut, joista sovellus koostuu, on ilmoitettava kuvion 3 rivien 15–76 mukaisesti. Komponenteille voidaan antaa erilaisia arvoja, kuten nimi, otsikko, teema ja kokoonpanon muuttumiseen vaikuttavia tekijöitä. Yksi aktiviteeteista on asetettava pääaktiviteetiksi kuvion 3 rivien 20–24 mukaisesti. Sovellusvaatimustiedostossa annetaan myös joidenkin rajapintojen käyttöön vaadittavat oikeudet. (Android Developers 2016c.) Kuvion 3 esimerkissä sovellukselle annetaan riveillä 5–8 käyttöoikeudet kalenteritietojen lukemiseen ja kirjoittamiseen sekä paikanuksen ja internet-yhteyden käyttöön.

TAULUKKO 1. Android-käyttöjärjestelmäversiot ja niiden laiteosuudet (Android Developers 2016i)

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.5%
4.1.x	Jelly Bean	16	8.8%
4.2.x		17	11.7%
4.3		18	3.4%
4.4	KitKat	19	35.5%
5.0	Lollipop	21	17.0%
5.1		22	17.1%
6.0	Marshmallow	23	1.2%

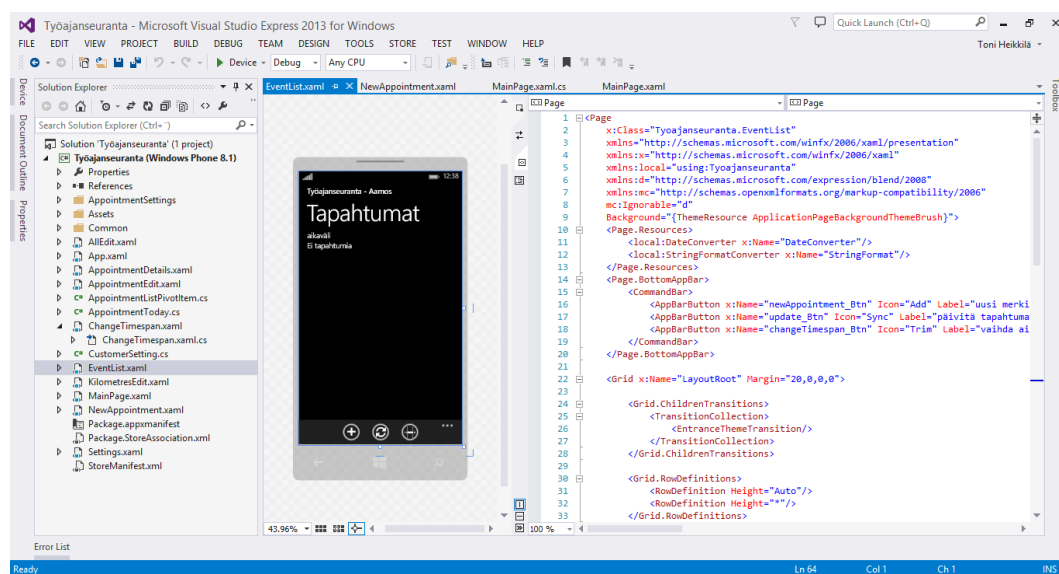
Android-käyttöjärjestelmään on julkaistu useita päivityksiä, jotka tuovat uusia ominaisuuksia. Uudet ominaisuudet eivät kuitenkaan toimi laitteissa, joissa ei ole uusimpia päivityksiä, joten kehittäjän on sovellusvaatimustiedostossa määritettävä vanhin Android-versio, jota sovellus tukee. (Android Developers 2016c.) Sovelluksen on kannattavaa tukea niin vanhoja Android-versioita kuin ominaisuuksien puolesta on mahdollista saavuttaakseen mahdollisimman paljon käyttäjiä. Taulukosta 1 nähdään, että esimerkiksi uusimmalla Marshmallow-nimisellä 6.0-käyttöjärjestelmäversiolla (API-taso 23) on vasta 1,2 prosentin osuus kaikista versioista. Sen sijaan yleisintä KitKat-nimistä versiota 4.4 (API-taso 19) tai sitä uudempaa käyttää noin 70 prosenttia kaikista Android-käyttäjistä. (Android Developers 2016i.)

Vanhimman tuettavan käyttöjärjestelmäversion lisäksi sovellusvaatimustiedostoon on vielä listattava esimerkiksi sijaintipalveluita varten tarvittavat kirjastot, joita sovellus mahdollisesti käyttää. Tiedostoon voidaan tehdä myös joitakin määrittäyksiä sovelluksen testausta varten, mutta ne on poistettava ennen sovelluksen julkaisua. (Android Developers 2016c.)

2.2 Windows Phone

2.2.1 Kehitysympäristö

Windows Phone -mobiilikäyttöjärjestelmälle sovelluksien kehittäminen tapahtuu Microsoftin kuvion 4 mukaista Visual Studio -sovelluskehitysympäristöä käyttäen. Kehitysympäristö mahdollistaa sovelluksen kehittämisen lisäksi sovelluksen testaamisen emulaattorilla ja fyysisellä laitteella. Laite on rekisteröitävä testilaitteeksi kehitysympäristöstä löytyvän rekisteröintityökalun avulla. Visual Studio -kehitysympäristöstä löytyy myös työkalut Windows Phone -kauppaan ladattavan asennuspaketin luomiseen. Visual Studio on saatavilla ilmaiseksi, mutta kaupalliseen käyttöön kehitettäviä sovelluksia varten on hankittava maksullinen versio. (Visual Studio 2016.)



KUVIO 4. Visual Studio -sovelluskehitysympäristö

Windows Phone -sovelluksia on mahdollista ohjelmoida C#-, C++- ja Visual Basic -ohjelmointikielillä. Kehitykseen suositellaan ensisijaisesti C#- ja Visual Basic -ohjelmointikieliä, mutta mikäli kehitetään pelejä tai muita graafisia sovelluksia, on C++ mahdollisesti parempi vaihtoehto. (Windows Dev Center 2016f.) Sovelluksia voidaan lisäksi luoda Windows Phone Sil-

verlight- tai Windows Phone Runtime -arkkitehtuureilla, mutta työssä keskitytään vain Runtime-arkkitehtuuriin. Sovelluksen ulkoasun luomisessa käytetään XAML-merkintäkieltä. (Microsoft Virtual Academy 2014a.)

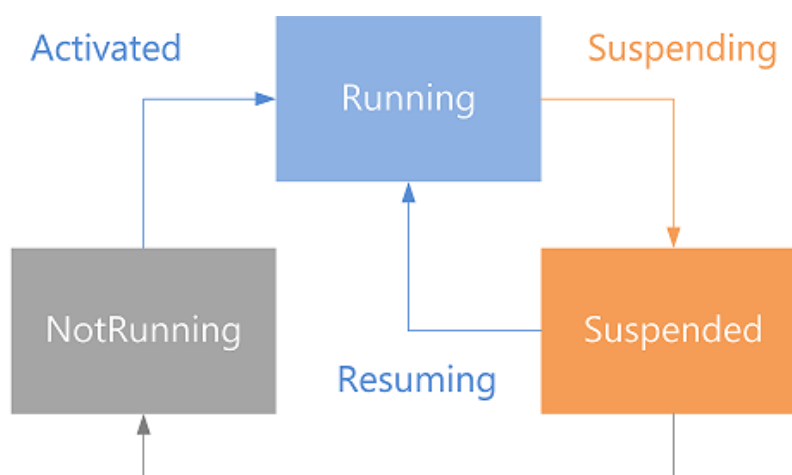
2.2.2 Sivut

Kaikki Windows Phone -sovelluksen näkymät sijaitsevat yhdessä kehyksessä. Windows Phone -kehityksessä käyttäjälle esitettäviä näkymiä kutsutaan sivuiksi. Kehys, jossa sovelluksen sivut sijaitsevat, huolehtii siirtymisestä sivulta toiselle. Kehyksellä on edellisistä sivuista pino, jonka päälimmäisenä on edellinen näkyvissä ollut sivu. Kehys luodaan sovellusta käynnistettäessä Application-luokassa, jossa määritetään myös sivu, joka näkyy käyttäjälle ensimmäisenä, kun sovellus on käynnistetty. Sivujen toiminnan kannalta tärkeitä ovat OnNavigatedTo- ja OnNavigatedFrom-metodit, joiden avulla suoritetaan tarvittavia toimintoja sivulle siirryttäessä ja sivulta poistuttaessa. (Microsoft Virtual Academy 2014c.)

Windows Phone -kehityksessä sivujen käyttöliittymä koostuu XAML-merkintäkielellä määritetyistä elementeistä. Elementtien avulla sivuille voidaan luoda erilaisia komponentteja, kuten painikkeita, tekstikenttiä ja graafista sisältöä. Komponenteilla on erilaisia ominaisuuksia, kuten sijainti, koko ja väri, niiden tarkempaa määrittämistä varten. Komponenttien luominen ja niiden määreiden asettaminen on mahdollista tehdä myös ohjelmallisesti ohjelmakoodin avulla. Komponentit asetellaan käyttöliittymän pääsääntöisesti joko pinomaisesti tai ruudukkomaisesti. Pinomaisessa asettelussa komponentit järjestetään pinoon joko vertikaalisesti tai horisontaalisesti. Ruudukkomaisessa asettelussa komponenteille luodaan ruudukko, jonka jokaiselle riville ja sarakkeelle määritetään korkeus ja leveys ruutuihin asetettavien komponenttien tarvitseman tilan mukaisesti. (Microsoft Virtual Academy 2014b.)

2.2.3 Sovelluksen elinkaari

Windows Phone -mobiilikäyttöjärjestelmässä suoritetaan vain yhtä sovel-
lusta kerrallaan, ellei sovellukseen ole luotu tausta-ajoa hyödyntäviä omi-
naisuuksia. Käyttäjän poistuessa sovelluksesta sen tila muuttuu joko kes-
keytetyksi tai lopetetuksi, riippuen tavasta, jolla käyttäjä sovelluksesta
poistuu. (Windows Dev Center 2016c.)



KUVIO 5. Sovelluksen elinkaari (Windows Dev Center 2016c)

Käyttäjän avatessa sovelluksen se voi olla joko keskeytetty- tai lopetettu-
tilassa. Mikäli sovellus on kuvion 5 mukaisesti NotRunning- eli lopetettu-
tilassa, on sovellus käynnistettävä uudelleen. Normaalisti sovellus on Not-
Running-tilassa, kun se käynnistetään asentamisen jälkeen ensimmäistä
kertaa tai sen suorittaminen on edellisen käyttökerran jälkeen lopetettu
käyttäjän tai järjestelmän toimesta. Sovellusta uudelleen käynnistettäessä
käyttäjälle näytetään käynnistyskuva, jonka aikana suoritetaan Applica-
tion-luokan OnLaunched-metodi, joka varmistaa sovelluksen käyttöliitty-
män olevan valmiina näytettäväksi käyttäjälle. Sovelluksessa mahdollisesti
käytettävät edelliseen käyttökertaan liittyvät tiedot tulisi myös palauttaa
tässä vaiheessa OnActivated-metodissa. (Windows Dev Center 2016c.)

Sovelluksen tila muuttuu keskeytetyksi käyttäjän poistuessa sovelluksesta.
Samalla suoritetaan myös kuvion 5 mukaisesti Suspended-tapahtuma,

jossa voidaan esimerkiksi tallettaa tarvittavat sovelluksen tiedot, jotta ne voidaan palauttaa, mikäli sovellus on jouduttu lopettamaan ennen kuin käyttäjä seuraavan kerran palaa sovellukseen. Käyttöjärjestelmä pyrkii pitämään mahdollisimman monta sovellusta keskeytettynä, mutta mikäli järjestelmän resurssit loppuvat, on sen vapautettava resursseja lopettamalla sovelluksia. Sovellus voidaan lopettaa myös käyttäjän toimesta sovellusvalitsimen kautta. Sovellukseen on mahdollista toteuttaa tausta-ajoa hyödyntäviä ominaisuuksia, joiden avulla sovellus voi suorittaa toimintoja sen ollessa keskeytettynä tai jopa lopetettuna. Keskeytetty-tilassa oleva sovellus pysyy käyttöjärjestelmän muistissa, joten käyttäjän palatessa sovellukseen ovat sen tiedot pysyneet muuttumattomina. (Windows Dev Center 2016c.)

Käyttäjän palatessa keskeytettynä olevaan sovellukseen sen tila muuttuu aktiiviseksi. Tällöin suoritetaan myös kuviossa 5 esiintyvä Resuming-tapahtuma, jossa voidaan päivittää esimerkiksi käyttöliittymässä mahdollisesti olevia verkosta haettavia sisältöjä, jotka voivat muuttua sovelluksen ollessa keskeytettynä. (Windows Dev Center 2016c.)

2.2.4 Sovellusvaatimustiedosto

Windows Phone -kehityksessä sovelluksen vaatimukset määritellään package.appxmanifest-nimisessä XML-tiedostossa. Se sisältää sovelluksen kauppaan lähettämiseen tarvittavia ominaisuuksia ja asetuksia. Visual Studio -sovelluskehitysympäristö tarjoaa kuvion 6 mukaisen graafisen työkalun tiedoston muokkaamiseen, mutta sitä on mahdollista muokata myös normaalin tekstimuotoisen XML-tiedoston tavoin. (Microsoft Developer Network 2016a.)

Application Visual Assets Requirements Capabilities Declarations Content URIs Packaging

Use this page to set the properties that identify and describe your app.

Display name: Aamos TAS työajanseuranta

Entry point: Työajanseuranta.App

Default language: fi-FI [More information](#)

Description: Työajanseuranta

Supported rotations: An optional setting that indicates the app's orientation preferences.

☐ Landscape ☒ Portrait ☐ Landscape-flipped

SD cards: ☒ Prevent installation to SD cards

Notifications:

Toast capable: (not set)

Lock screen notifications: (not set)

Tile Update:

Updates the app tile by periodically polling a URI. The URI template can contain "{language}" and "{region}" tokens that will be replaced at runtime to generate the URI to poll.

KUVIO 6. Sovellusvaatimustiedoston määrittäminen graafisella työkalulla

Sovelluksen yleisiä tietoja määritetään Application-välilehdellä. Määritettävänä ovat esimerkiksi sovelluksen nimi, oletuskieli sekä ilmoitusten käyttö. Lisäksi on määritettävä aloituskohta, josta sovelluksen käynnistäminen aloitetaan. Aloituskohta on oletuksena Application-luokka. Visuaaliset sisällöt, kuten erikokoiset sovelluskuvakkeet ja käynnistyskuvat, asetetaan Visual Assets -välilehdellä. Mikäli sovellus vaatii laitteelta joitain tiettyjä ominaisuuksia toimiakseen, määritetään ne Requirements-välilehdellä. Tällaisia ominaisuuksia ovat gyroskooppi, jolla määritetään laitteen asento, kompassi, muun muassa lähimaksun ja laitteen etäluvun mahdollistava NFC-siru sekä etu- ja takakamera. (Microsoft Virtual Academy 2014e; Microsoft Virtual Academy 2014f.)

Sovelluksessa käytettävät ja käyttöoikeuden vaativat toiminnot tulee määrittää Capabilities-välilehden alla. Käyttöoikeuden vaativia toimintoja ovat esimerkiksi paikannuspalveluiden, kalenterimerkintöjen, internet-yhteyden ja mikrofoniin käyttö. Declaration-välilehdellä sovellus voidaan määrittää olevan muiden sovellusten käytettävissä joihinkin toimintoihin, kuten tiedostojen tallentamiseen tai avaamiseen. Saman välilehden alla esitellään ja toteutetaan myös sovelluksessa olevat tausta-ajoa hyödyntävät toiminnot. Mikäli sovelluksessa avataan ja näytetään internet-sivuja, voidaan

Content URIs -välilehdellä määrittää niiden sivujen osoitteet, joille halutaan sallia tiedon välittäminen sovellukseen. Lopuksi Packaging-välilehdellä sovellukselle määritetään sovelluksen yksilöivä nimi. Tämän lisäksi sovellukselle syötetään versionumero ja julkaisijan nimi sekä määritetään asennuspaketin luomiseen liittyviä ominaisuuksia. (Microsoft Virtual Academy 2014f.) Packaging-välilehdellä olevat tiedot voidaan määrittää myös asennuspaketin luomisen yhteydessä (Microsoft Virtual Academy 2014e).

3 KALENTERIRAJAPINTOJEN KÄYTTÖ

3.1 Android

3.1.1 Kalenterimerkintöjen hakeminen

Kalenterimerkintöjen hakemiseen käytetään ContentResolver-luokkaa, jonka avulla päästään turvallisesti käsiksi puhelimen tietoihin, eli tässä tapauksessa kalenterimerkintöihin (Android Developers 2016e). Kalenterimerkinnöistä on mahdollista lukea erilaisia tietoja, kuten merkinnän nimi, kuvaus sekä aloitus- ja lopetusaika. ContentResolver-luokasta käytetään query-metodia, joka palauttaa Cursor-tyyppisen olion, joka sisältää kaikki löydetty kalenterimerkinnät. (Android Developers 2016d.)

```

180      // Haetaan merkinnät kalenterista
181      Cursor cur;
182      ContentResolver cr = this.getActivity().getContentResolver();
183      Uri uri = Events.CONTENT_URI;
184
185      // Määritetään millä ehdoilla haetaan merkintöjä
186      String selection = "( " + Events.DTSTART + " >= " + savedStartDate + " ) " +
187          "AND ( " + Events.DTSTART + " <= " + savedEndDate + " ) " +
188          "AND ( " + Events.TITLE + " = " + appointmentType + " ) " +
189          "AND ( " + Events.CALENDAR_ID + " = " + savedCalID + " )";
190
191
192      // Vaihdetaan listan järjestys (uusin päivä listan ensimmäiseksi)
193      String orderBy = Events.DTSTART + " DESC";
194
195      // Haetaan merkinnät kalenterista
196      cur = cr.query(uri, APPOINTMENT_PROJECTION, selection, null, orderBy);

```

KUVIO 7. Kalenterimerkintöjen hakeminen

Query-metodin ensimmäisenä parametrina määritetään Uri-muuttujatyypin avulla, minkä tyyppistä tietoa haetaan, eli tässä tapauksessa kalenterimerkintöjä, kuten kuviossa 7 rivillä 183 on määritetty. Toiseksi parametriksi asetetaan kuvion 8 mukainen merkkijonotaulukko niistä tiedoista, joita merkinnöistä halutaan hakea. Kuviossa 8 halutuiksi tiedoiksi on määritetty kalenterimerkinnän yksilöivä tunniste, merkinnän otsikko, aloitusaika, sijainti, kuvaus ja yksilöivä tunniste kalenterille, jolle merkintä kuuluu. (Android Developers 2016f; Android Developers 2016d.)

```

47     public static final String[] APPOINTMENT_PROJECTION = new String[]{
48         CalendarContract.Events._ID,                // 0
49         CalendarContract.Events.TITLE,              // 1
50         CalendarContract.Events.DTSTART,            // 2
51         CalendarContract.Events.EVENT_LOCATION,      // 3
52         CalendarContract.Events.DESCRPTION,         // 4
53         CalendarContract.Events.CALENDAR_ID         // 5
54     };

```

KUVIO 8. Kalenterimerkinnöistä haettavat tiedot

Query-metodin kolmantena parametrina on mahdollista määrittää hakuun ehtoja. Kuvion 7 tapauksessa riveillä 186–189 on asetettu merkkijonoon ehdot, jotka määrittävät, miltä aikaväliltä kalenterimerkinnät haetaan, mikä on haettavien merkintöjen otsikko sekä mistä kalenterista merkinnät haetaan. Neljänneksi parametriksi asetetaan merkkijonotaulukko, johon asetetaan ne arvot, joita halutaan käyttää ehtolauseessa mahdollisten kysymysmerkkien kohdalla. Kuvion 7 tapauksessa kaikki arvot on määritetty kolmanneksi parametriksi asetetussa ehtolauseessa, joten tämä parametri on asetettu tyhjäksi (null). Viimeiseksi parametriksi voi asettaa järjestyksen, jossa haluaa merkintöjen palautettavan. (Android Developers 2016d; Android Developers 2016f.) Kuviossa 7 rivillä 193 asetetaan järjestys aloitusajan mukaan uusimmasta vanhimpaan.

Kalenterimerkintöjen hakemisen jälkeen kaikki haetut merkinnät voidaan esimerkiksi käydä yksitellen läpi Cursor-luokan moveToNext-metodin avulla, kuten kuviossa 9, ja käsitellä niitä sovelluksen tarvitsemalla tavalla. Jokaisella kalenterimerkinnän tiedolla on oma tietotyyppinsä, ja tiedot on järjestetty indeksien mukaan samaan järjestykseen, johon ne oli asetettu query-metodiin toiseksi parametriksi määritetyssä merkkijonotaulukossa. (Android Developers 2016d.)

```

203     long apptID = 0;
204     String title = null;
205     long dateInMillis = 0;
206     String location = null;
207     String description = null;
208     long calID = 0;
209
210     // Käydään lävitse haetut merkinnät
211     while (cur.moveToNext()) {
212
213         // Asetetaan merkinnän tiedot muuttujiin
214         apptID = cur.getLong(0);
215         title = cur.getString(1);
216         dateInMillis = cur.getLong(2);
217         Date date = new Date(dateInMillis);
218         location = cur.getString(3);
219         description = cur.getString(4);
220         calID = cur.getLong(5);
221
222         // Talletetaan löydetty merkintä...
223     }

```

KUVIO 9. Haettujen kalenterimerkintöjen läpikäynti

3.1.2 Kalenterimerkinnän luominen

Uuden kalenterimerkinnän lisäämisessä käytetään merkintöjen hakemisen mukaisesti ContentResolver-luokkaa. Lisääminen tapahtuu käyttämällä luokan insert-metodia, kuten kuviossa 10 rivillä 163. Metodin ensimmäiseksi parametriksi asetetaan tieto siitä, millaista tietoa ollaan käsittelemässä, kuten tehtiin kalenterimerkintöjä haettaessakin. Toisessa parametrissa määritetään ContentValues-tyyppisen olion avulla tiedot, jotka kalenterimerkintään halutaan tallettaa. (Android Developers 2016f.) ContentValues-luokan put-metodia käyttäen lisätään haluttuja määreitä merkintään. Metodin ensimmäiseksi parametriksi asetetaan haluttu kalenterimerkinnän kenttä ja toiseksi parametriksi kenttään asetettavaksi haluttu arvo kuvion 10 riveillä 157–162 osoittamalla tavalla, jossa merkinnälle asetetaan alkamis- ja päättymisaika, nimi, kuvaus, kalenteri ja aikavyöhyke. (Android Developers 2016d; Android Developers 2016g.)

```

145     long calID = 3;
146     long startMillis = 0;
147     long endMillis = 0;
148     Calendar beginTime = Calendar.getInstance();
149     beginTime.set(2012, 9, 14, 7, 30);
150     startMillis = beginTime.getTimeInMillis();
151     Calendar endTime = Calendar.getInstance();
152     endTime.set(2012, 9, 14, 8, 45);
153     endMillis = endTime.getTimeInMillis();
154
155     ContentResolver cr = getContentResolver();
156     ContentValues values = new ContentValues();
157     values.put(Events.DTSTART, startMillis);
158     values.put(Events.DTEND, endMillis);
159     values.put(Events.TITLE, "Jazzercise");
160     values.put(Events.DESCRPTION, "Group workout");
161     values.put(Events.CALENDAR_ID, calID);
162     values.put(Events.EVENT_TIMEZONE, "America/Los_Angeles");
163     Uri uri = cr.insert(Events.CONTENT_URI, values);

```

KUVIO 10. Uuden kalenterimerkinnän lisääminen (Android Developers 2016d)

Uuden kalenterimerkinnän lisäämisessä on otettava huomioon, että kalenteri, johon merkintä lisätään, aloitusaika ja merkinnän aikavälyhyke ovat pakollisia tietoja, joita ilman kalenterimerkinnän lisääminen ei onnistu. Lisäksi merkinnöissä, jotka eivät ole toistuvia, täytyy olla päättymisaika, ja sen sijaan toistuvista merkinnöistä on löydyttävä merkinnän kesto ja tieto siitä, minä päivinä merkintä toistuu. (Android Developers 2016d.)

3.1.3 Kalenterimerkinnän päivittäminen

Kalenterimerkinnän päivittämisessäkin käytetään hyväksi Content-Resolver-luokan tarjoamia metodeita. Päivittäminen tapahtuu update-metodilla, joka palauttaa lukuarvon siitä, kuinka monta merkintää päivitettiin. Ensimmäiseksi parametriksi asetetaan kalenterimerkintöjen hakemisen ja lisäämisen tapaan tieto siitä, millaista tietoa ollaan päivittämässä. Sen lisäksi samaan Uri-tyyppiseen muuttujaan lisätään kuvion 11 rivin 118 mukaisesti kalenterimerkinnän yksilöivä tunniste, jonka avulla määritetään, mitä yksittäistä merkintää halutaan päivittää. (Android Developers 2016d.)

```

113 // Päivitetään merkintä
114 cr = getContentResolver();
115 values = new ContentValues();
116 updateUri = null;
117 values.put(CalendarContract.Events.EVENT_LOCATION, "Myrskylä");
118 updateUri = ContentUris.withAppendedId(CalendarContract.Events.CONTENT_URI, apptID);
119 updatedRows = cr.update(updateUri, values, null, null);

```

KUVIO 11. Kalenterimerkinnän päivittäminen

Toiseksi parametriksi määritetään ContentValues-luokan avulla kentät ja arvot, jotka halutaan päivittää, kuten tehtiin kalenterimerkinnän lisäämisessä lisättävien tietojen kohdalla. Kaksi viimeistä parametria toimivat samalla lailla kuin ehtolauseetta määritettäessä kalenterimerkintöjä haettaessa. (Android Developers 2016d; Android Developers 2016f.)

3.1.4 Kalenterimerkinnän poistaminen

Kalenterimerkinnän poistaminen tapahtuu ContentResolver-luokan delete-metodin avulla, toimien lähes identtisesti merkinnän päivittämisen kanssa. Delete-metodin ensimmäinen parametri toimii samalla lailla kuin update-metodissa. Toiseen parametriin ei delete-metodin yhteydessä luonnollisestikaan määritetä merkinnälle mitään uusia arvoja, vaan kaksi seuraavaa ja viimeistä parametria määritetään samalla tavalla kuin update-metodin kaksi viimeistä parametria, niiden ollessa siis poisto-operaation mahdollisia ehtomäärittelyjä. Delete-metodi palauttaa tiedon siitä, kuinka monta merkintää kalenterista poistettiin. (Android Developers 2016d.)

3.1.5 Kalenteritietojen hallinta kalenterisovelluksen avulla

Androidin kalenteritietoja on mahdollista käsitellä myös siten, että kalenterimerkinnän lisäämistä, muokkaamista ja tarkastelua varten avataan puhelimesta oleva kalenterisovellus ja suoritetaan halutut toiminnot siellä. Tässä tapauksessa omalle sovellukselleen ei tarvitse ohjelmoida kyseisiä toimintoja. Erillisen kalenterisovelluksen avulla kalenteritietojen käsittely tapahtuu Intent-luokan avulla. (Android Developers 2016d.)

Kalenterimerkinnän lisääminen erillisen kalenterisovelluksen avulla tapahtuu luomalla Intent-luokasta olio, jolle alustettaessa annetaan parametrina tieto siitä, että ollaan luomassa lisäystoimintoa, kuten kuviossa 12 on rivillä 218 tehty. Tämän lisäksi setData-metodilla määritetään, että kyseessä on kalenterimerkinnän lisääminen. Avattavan lisäysnäköymän kenttiin on mahdollista asettaa tietoa jo ennen kuin käyttäjä on niitä asettanut. Tämä tapahtuu kuvion 12 rivien 220–226 mukaisesti käyttämällä putExtra-metodia, jota käytetään samoin kuin ContentValues-luokan put-metodia, jota käytettiin, kun lisättiin ja muokattiin merkintöjä ohjelmallisesti. Lisäysnäköymä käynnistetään kuvion 12 rivin 227 mukaisesti startActivity-metodilla, jonka parametrina on aiemmin luotu Intent-olio. (Android Developers 2016d.)

```

214     Calendar beginTime = Calendar.getInstance();
215     beginTime.set(2012, 0, 19, 7, 30);
216     Calendar endTime = Calendar.getInstance();
217     endTime.set(2012, 0, 19, 8, 30);
218     Intent intent = new Intent(Intent.ACTION_INSERT)
219         .setData(Events.CONTENT_URI)
220         .putExtra(CalendarContract.EXTRA_EVENT_BEGIN_TIME, beginTime.getTimeInMillis())
221         .putExtra(CalendarContract.EXTRA_EVENT_END_TIME, endTime.getTimeInMillis())
222         .putExtra(Events.TITLE, "Yoga")
223         .putExtra(Events.DESCRPTION, "Group class")
224         .putExtra(Events.EVENT_LOCATION, "The gym")
225         .putExtra(Events.AVAILABILITY, Events.AVAILABILITY_BUSY)
226         .putExtra(Intent.EXTRA_EMAIL, "rowan@example.com,trevor@example.com");
227     startActivity(intent);

```

KUVIO 12. Kalenterimerkinnän lisääminen Intent-luokan avulla (Android Developers 2016d)

Kalenterimerkinnän päivittäminen erillisessä näkymässä tapahtuu hyvin samalla tavalla kuin lisääminenkin. Intent-oliota luodessa sen alustus-metodin parametriksi annetaan tieto, että luodaan päivitysnäköymä. Lisäksi setData-metodiin on määritettävä päivitettävän tiedon tyypin lisäksi päivitettävän kalenterimerkinnän yksilöivä tunniste kuvion 13 rivin 83 mukaisesti. Myös merkintää päivitettäessä on mahdollista määrittää tietyille kentille jokin arvo putExtra-metodin avulla, kuten uutta merkintää lisättäessäkin, ja lopuksi lisäysnäköymä käynnistetään jälleen startActivity-metodin avulla. (Android Developers 2016d.)


```

83      Uri uri = ContentUris.withAppendedId(Events.CONTENT_URI, eventID);
84      Intent intent = new Intent(Intent.ACTION_EDIT)
85          .setData(uri)
86          .putExtra(Events.TITLE, "My New Title");
87      startActivity(intent);

```

KUVIO 13. Kalenterimerkinnän päivittäminen Intent-luokan avulla (Android Developers 2016d)

Kalenterimerkinnän katsominen Intent-luokan avulla onnistuu lähes samoin kuin päivittäminen. Ainoastaan Intent-oliota luotaessa alustus-metodin parametriksi vaihdetaan merkinnän katselunäkymä (ACTION_VIEW), eikä kenttien arvoja pysty muokkaamaan putExtra-metodilla, koska tarkoituksena on vain katsoa merkinnän tietoja. (Android Developers 2016d.)

3.1.6 Kalenterien hakeminen

Kalenterien hakeminen tapahtuu hyvin samalla tavalla kuin itse kalenterimerkintöjenkin hakeminen. Kalentereita haettaessa käytetään niin ContentResolver-luokan query-metodia kuin Cursor-oliotakin, aivan kuten merkintöjä haettaessa. (Android Developers 2016d.)

```

99      // Haetaan kalenterit
100     Cursor cur = null;
101     ContentResolver cr = getContentResolver();
102     Uri uri = Calendars.CONTENT_URI;
103
104     cur = cr.query(uri, CALENDAR_PROJECTION, null, null, null);

```

KUVIO 14. Kalenterien hakeminen

Query-metodi toimii kalentereita haettaessa kuvion 14 mukaisesti. Ensimmäiseksi parametriksi määritetään tieto siitä, että haetaan kalentereita ja toiseen parametriin asetetaan merkkijonotaulukko, joka kertoo, mitä tietoja kalentereista halutaan. Kalentereista on mahdollista hakea esimerkiksi niiden yksilöivä tunniste, nimi ja omistaja. Kolmella viimeisellä parametrilla

on mahdollista määrittää hakuehdot ja järjestys, jossa kalenterit palaute-
taan Cursor-olioon. Hakemisen jälkeen kalenterien tiedot ovat sovelluksen
käytävissä. (Android Developers 2016d.)

3.2 Windows Phone

3.2.1 Kalenterimerkintöjen hakeminen

Kalenterimerkintöjen hakemiseen käytetään AppointmentCalendar-luokan
FindAppointmentsAsync-metodia. Metodin avulla saadaan lista haetuista
kalenterimerkinnöistä. Metodin kutsumista varten on ensin oltava Appoint-
mentCalendar-tyyppinen olio, jolta kyseinen metodi löytyy, koska tämä
määrittää, mistä kalenterista kalenterimerkintöjä haetaan. FindAppoint-
mentsAsync-metodin ensimmäiseksi parametriksi asetetaan DateTime-
Offset-muuttujatyypinen aloitusaika, josta lähtien kalenterimerkintöjä ha-
lutaan hakea. Toisen parametrin avulla määritetään aika, kuinka paljon
eteenpäin ensimmäiseksi parametriksi asetetusta aloitusajasta kalenteri-
merkintöjä haetaan. (Microsoft Virtual Academy 2014d.)

```

125 // Määritetään, mitkä tiedot kalenterimerkinnästä haetaan
126 FindAppointmentsOptions options = new FindAppointmentsOptions();
127 options.FetchProperties.Add(AppointmentProperties.Subject);
128 options.FetchProperties.Add(AppointmentProperties.StartTime);
129 options.FetchProperties.Add(AppointmentProperties.Location);
130 options.FetchProperties.Add(AppointmentProperties.Details);
131 options.FetchProperties.Add(AppointmentProperties.Duration);
132
133 // Haetaan kaikki kalenterimerkinnät valitulta aikaväliltä
134 var appointments = await App.apptCalendar.FindAppointmentsAsync(App.start,
135     new TimeSpan(App.daysDuration, 0, 0, 0), options);

```

KUVIO 15. Kalenterimerkintöjen hakeminen

FindAppointmentsAsync-metodin viimeisellä parametrilla määritetään
FindAppointmentsOptions-luokan olion avulla tiedot, jotka kalenterimerkin-
nöistä haetaan. Haettavia tietoja lisätään FindAppointmentsOptions-luo-
kan FetchProperties-ominaisuuden Add-metodin avulla kuvion 15 rivien
127–131 mukaisesti, jolloin merkinnöistä haetaan aihe, aloitusaika, sijainti,

lisätiedot ja kesto (Windows Dev Center 2016i). FindAppointmentsAsync-metodin palauttama lista sisältää kalenterimerkinnät Appointment-olioina, joita voi niiden hakemisen jälkeen käyttää sovelluksessa tarvittavalla tavalla. (Microsoft Virtual Academy 2014d.)

3.2.2 Kalenterimerkinnän luominen

Kalenterimerkinnän luomisessa käytetään hyödyksi AppointmentManager-luokan ShowAddAppointmentAsync-metodia. ShowAddAppointmentAsync-metodin lisäksi merkinnän luomisessa hyödynnetään Appointment-luokkaa, josta luotava olio on myös käytettävän metodin ensimmäinen parametri. Appointment-luokasta löytyy kaikki kalenterimerkinnän ominaisuudet, kuten merkinnän aloitusaika, aihe, kuvaus, sijainti ja kesto, jotka on asetettu olioon kuvion 16 riveillä 371–376. (Microsoft Virtual Academy 2014d; Windows Dev Center 2016j.)

```

363 // Luodaan uusi kalenterimerkintä
364 Appointment appt = new Appointment();
365
366 var date = (DateTimeOffset)DateTime.Now;
367 var time = (TimeSpan)new TimeSpan(12, 0, 0);
368 var timeZoneOffset = TimeZoneInfo.Local.GetUtcOffset(DateTime.Now);
369 var startTime = new DateTimeOffset(date.Year, date.Month, date.Day, time.Hours,
370     time.Minutes, 0, timeZoneOffset);
371 appt.StartTime = startTime;
372
373 appt.Subject = "Aihe";
374 appt.Details = "Lisätiedot";
375 appt.Location = "Sijainti";
376 appt.Duration = TimeSpan.FromMinutes(30);
377 String apptID = await AppointmentManager.ShowAddAppointmentAsync(appt, new Rect());

```

KUVIO 16. Kalenterimerkinnän lisääminen

Toiseksi parametriksi on asetettava Rect-olio, jolla voisi määrittää uuden kalenterimerkinnän lisäysnäkyvän sijainnin ja koon. Tällä ei ole kuitenkaan vaikutusta mobiilisovellusta luotaessa, vaan se on tarkoitettu ainoastaan työpöytäsovelluksia varten. ShowAddAppointmentAsync-metodi palauttaa luodun kalenterimerkinnän yksilöivän tunnisteen, mikäli merkinnän luonti onnistui. (Microsoft Virtual Academy 2014d.)

3.2.3 Kalenterimerkinnän päivittäminen

Kalenterimerkinnän päivittämiseen käytetään AppointmentStore-luokkaa. Ensin luokasta luodaan olio määrittämällä siihen AppointmentManager-luokan RequestStoreAsync-metodin avulla kalenterimerkinnät luku- ja kirjoitusoikeuksilla kuvion 17 rivien 152 ja 153 mukaisesti. (Microsoft Virtual Academy 2014d; Windows Dev Center 2016e.)

```

147 // Muokataan kalenterimerkintää
148 App.appt.Location = "Myrskylä";
149 App.appt.Duration = TimeSpan.FromHours(1.5);
150 App.appt.Details = "Uusi sijainti ja kesto";
151
152 AppointmentStore apptStore =
153     await AppointmentManager.RequestStoreAsync(AppointmentStoreAccessType.AppCalendarsReadWrite);
154
155 String apptID =
156     await apptStore.ShowReplaceAppointmentAsync(App.appt.LocalId, App.appt, new Rect());

```

KUVIO 17. Kalenterimerkinnän päivittäminen

AppointmentStore-olion määrittämisen jälkeen kalenterimerkinnän päivittäminen onnistuu AppointmentStore-luokan ShowReplaceAppointmentAsync-metodin avulla. Metodin ensimmäiseksi parametriksi annetaan päivitettävän merkinnän paikallinen yksilöivä tunniste, joka on yksi kalenterimerkinnän Appointment-olion ominaisuuksista. Toiseksi parametriksi määritetään itse Appointment-olio. Kalenterimerkintään tehtävät muutokset määritetään etukäteen kyseiseen olioon, kuten kuviossa 17 on tehty riveillä 148–150, jossa on muutettu merkinnän sijaintia, kestoja ja lisätietoja. Viimeiseksi parametriksi asetetaan uuden merkinnän lisäämisen tapaisesti Rect-olio. ShowReplaceAppointmentAsync-metodi palauttaa onnistuessaan päivitetyn kalenterimerkinnän yksilöivän tunnisteen. (Windows Dev Center 2016o.)

3.2.4 Kalenterimerkinnän poistaminen

Kalenterimerkinnän poistaminen tapahtuu päivittämisen tavoin AppointmentStore-luokkaa käyttäen. Luokasta luodaan olio samalla tavalla kuin

kuviossa 17 riveillä 152 ja 153 merkinnän päivittämisen yhteydessä. Merkintää poistettaessa AppointmentStore-luokasta käytetään ShowRemoveAppointmentAsync-metodia. (Windows Dev Center 2016e.) Metodilla on kaksi parametria, joista ensimmäisessä määritetään poistettavan kalenterimerkinnän yksilöivä tunniste. Toiseksi parametriksi asetetaan lisäys- ja päivitystoiminnoista tuttu Rect-olio. ShowRemoveAppointmentAsync-metodi palauttaa Boolean-tyyppisen totuusarvon poistamisen onnistumisesta. (Microsoft Virtual Academy 2014d.)

3.2.5 Kalenterien hakeminen

Kalenterit haetaan puhelimesta hyödyntäen AppointmentStore- ja AppointmentManager-luokkia kuvion 18 mukaisesti. Ensin AppointmentManager-luokan RequestStoreAsync-metodilla haetaan kalenterimerkinnät AppointmentStore-olioon. AppointmentStore-olion FindAppointmentCalendarsAsync-metodin avulla haetaan lopuksi kaikki kalenterit. Kalenterien hakemisen jälkeen niitä voi käyttää sovelluksen tarvitsemalla tavalla ja hakea esimerkiksi kalenteriin lisätyt merkinnät. (Microsoft Virtual Academy 2014d.)

```
136 // Haetaan kaikki kalenterit
137 AppointmentStore apptStore =
138     await AppointmentManager.RequestStoreAsync(AppointmentStoreAccessType.AllCalendarsReadOnly);
139
140 var apptCalendars = await apptStore.FindAppointmentCalendarsAsync();
```

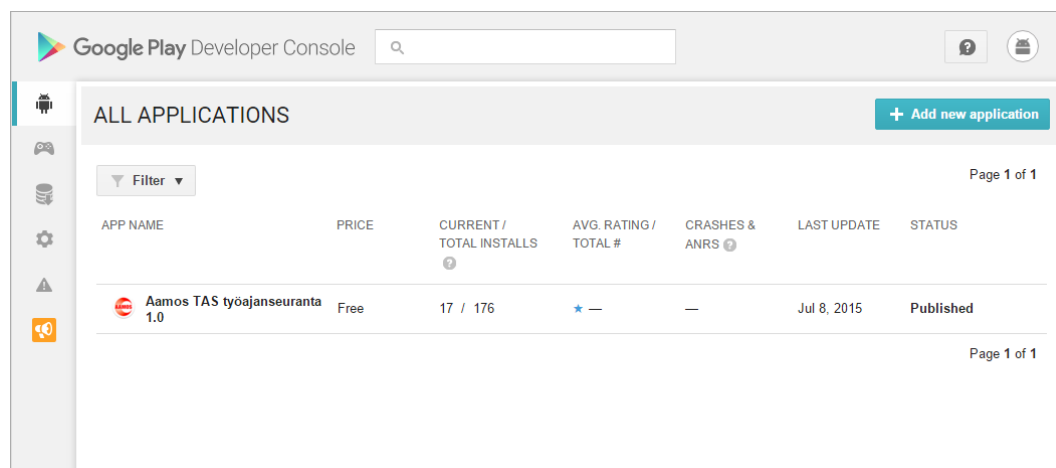
KUVIO 18. Kalenterien hakeminen

4 JULKAISEMINEN SOVELLUSKAUPPAAN

4.1 Google Play -kauppa

4.1.1 Kehittäjätilin luonti

Mobiilisovelluksien Google Play -kauppaan julkaisua varten on ensin luotava Google Play -kehittäjätili. Kehittäjätilin voi liittää jo olemassa olevaan Google-tiliin tai vaihtoehtoisesti voi luoda täysin uuden tilin. Kehittäjätilin luomisen yhteydessä on hyväksyttävä Google Playn kehittäjien jakelusopimus. Rekisteröityminen kehittäjäksi maksaa 25 dollaria eli noin 24 euroa tämänhetkisen valuuttakurssin mukaan (Suomen Pankki 2016). Kehittäjätiliä luotaessa on päätettävä muun muassa kehittäjänimi, joka näkyy Google Play -kaupan käyttäjille. Muita pakollisia tietoja ovat yhteyshenkilön sähköpostiosoite ja maksullisten sovellusten yhteydessä myös lähiosoite. (Google Play Developer Console Help 2016e.)



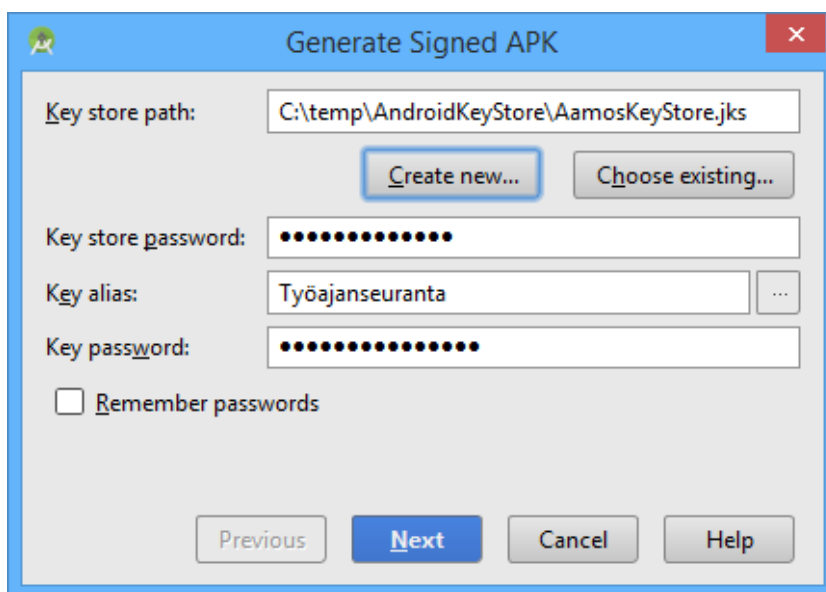
KUVIO 19. Google Play -kaupan kehittäjäkonsolin aloitusnäky

Kehittäjätilin luomisen jälkeen kehittäjällä on pääsy kuvion 19 mukaiseen Google Play -kaupan kehittäjäkonsoliin, jonka avulla sovelluksia on mahdollista julkaista kauppaan (Google Play Developer Console Help 2016e). Kehittäjätilin luoneen henkilön on ilman lisämaksua mahdollista lisätä omaan kehittäjäkonsoliinsa käyttöoikeuksia myös muille käyttäjille ja antaa

heille oikeudet julkaista sovelluksia tämän kehittäjätilin alla (Google Play Developer Console Help 2016a).

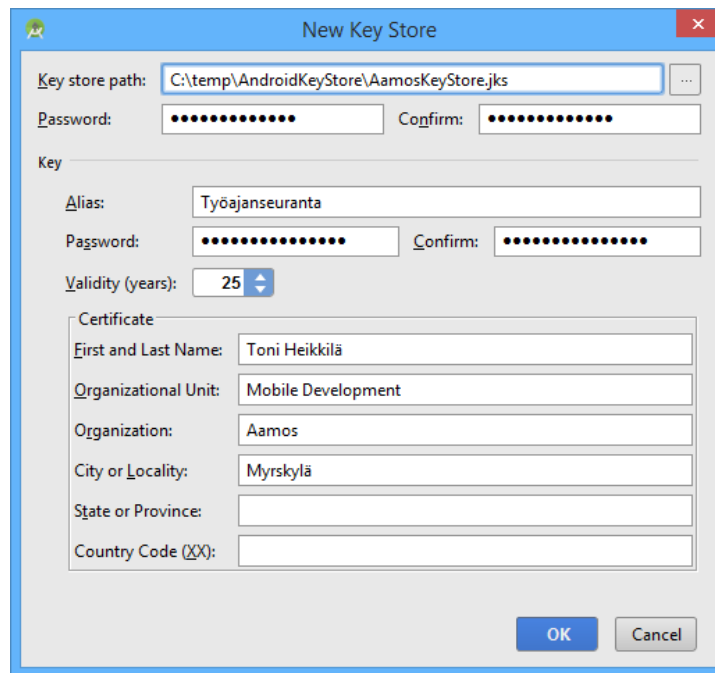
4.1.2 Sovelluksen asennustiedoston luonti

Google Play -kauppaan julkaistavasta Android-sovelluksesta on luotava allekirjoitettu apk-tiedostotyyppinen asennustiedosto. Asennustiedoston luominen tapahtuu Android Studio -sovelluskehitysympäristössä Generate Signed APK -työkalun avulla. Työkalun avauduttua esiin tulee kuvion 20 mukainen näkymä. Kehittäjän kaikilla sovelluksilla on yhteinen avainvarasto (key store), jossa säilötään avaimia, joilla sovelluksia allekirjoitetaan. Jos kehittäjällä ei vielä ole avainvarastoa, on se ensin luotava. (Android Developers 2016k.)



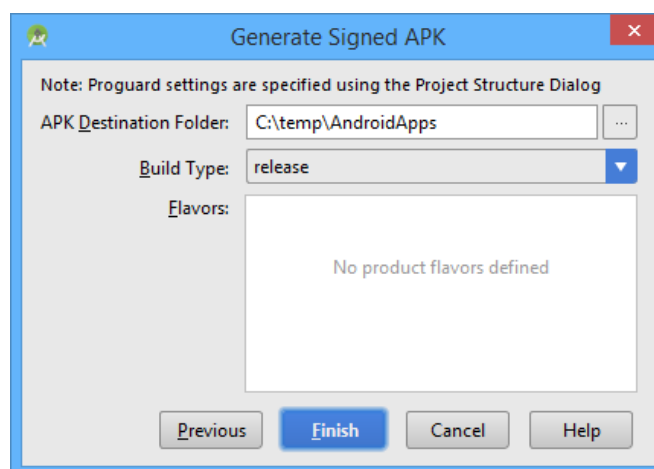
KUVIO 20. Asennustiedoston luomistyökalun avausnäkymä

Avainvarasto luodaan paikallisesti kehittäjän tietokoneelle haluttuun hake-
mistoon kuvion 21 mukaisesti ja se suojataan salasanalla. Samalla varas-
toon luodaan myös ensimmäinen sovelluskohtainen avain, jolle täytetään
vaaditut tiedot. Avaimen voimassaoloajaksi pitäisi määrittää vähintään 25
vuotta, jotta avain olisi voimassa koko sovelluksen elinkaaren ajan.
(Android Developers 2016k.)



KUVIO 21. Avainvaraston ja avaimen luominen

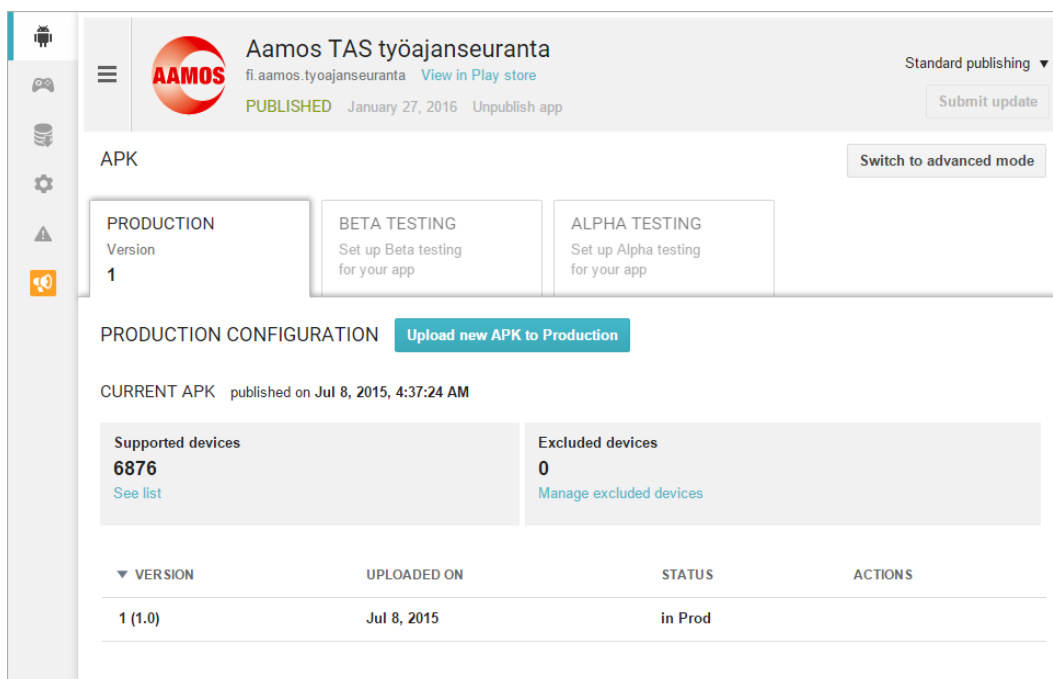
Avainvaraston ja avaimen luomisen jälkeen siirrytään kuvion 20 näkymän kautta asennustiedoston luomiseen, kun käytettävät avaimet on valittu. Kuvion 22 näkymässä viimeistellään asennustiedoston luominen määrittämällä hakemisto, jonne asennustiedosto halutaan luoda, ja valitsemalla koontiversioksi julkaisuversio. Allekirjoitetun apk-asennustiedoston luomisen jälkeen sovellus on valmiina kauppaan julkaisemista varten. (Android Developers 2016k.)



KUVIO 22. Asennustiedoston hakemiston ja koontiversion valinta

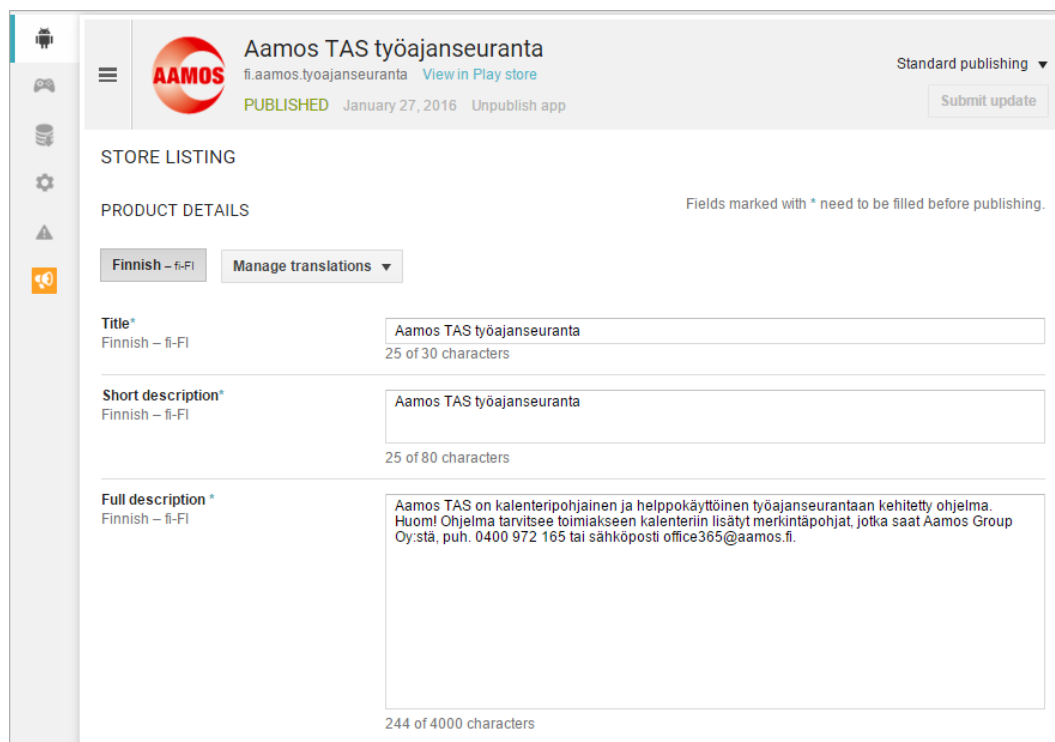
4.1.3 Sovelluksen julkaiseminen kauppaan

Android-sovelluksen julkaiseminen Google Play -kauppaan tapahtuu kehittäjäkonsolin kautta. Ensimmäisenä uutta sovellusta julkaistaessa kysytään sovelluksen oletuskieltä sekä nimeä, jonka käyttäjät näkevät Google Play -kaupassa. Tämän vaiheen jälkeen siirrytään lataamaan apk-asennustiedosto kehittäjäkonsoliin. (Google Play Developer Console Help 2016i.)



KUVIO 23. Android-asennustiedoston lataaminen kehittäjäkonsoliin

Asennustiedoston lataamisen kanssa samassa, kuvion 23 mukaisessa näkymässä voidaan sovellus asettaa myös alpha- tai betatestaukseen. Kehittäjä voi näiden ominaisuuksien avulla julkaista sovelluksen testattavaksi haluamalleen käyttäjäryhmälle ennen sovelluksen virallista julkaisua. Tällä tavoin kehittäjä voi korjata sovelluksessa testausvaiheessa mahdollisesti ilmenneet ongelmat tai muut testausryhmää vaivanneet asiat, kuten käyttöliittymän vaikeakäyttöisyys, jo ennen kuin sovellus on kaikkien saatavilla ja ladattavissa Google Play -kaupassa. (Google Play Developer Console Help 2016i.)



Aamos TAS työajanseuranta
fi.aamos.tyoajanseuranta [View in Play store](#)
PUBLISHED January 27, 2016 Unpublish app [Submit update](#)

STORE LISTING

PRODUCT DETAILS Fields marked with * need to be filled before publishing.

Finnish – fi-FI [Manage translations](#)

Title*
Finnish – fi-FI
25 of 30 characters

Short description*
Finnish – fi-FI
25 of 80 characters

Full description*
Finnish – fi-FI
244 of 4000 characters

KUVIO 24. Android-sovelluksen tuotetietojen täyttö

Asennustiedoston lataamisen jälkeen siirrytään täyttämään kuvion 24 mukaisesti sovelluksen tuotetiedot, jotka näkyvät käyttäjille Google Play -kaupassa. Tuotetietojen on tarkoitus olla mahdollisimman kuvaavat, jotta käyttäjä tietää, millainen sovellus on kyseessä. Sovelluksesta on mahdollista kirjoittaa kaksi eri kuvausta: lyhyempi, enimmillään 80 merkkiä pitkä kuvaus, joka näkyy ensimmäisenä sovelluksen tietosivulla, sekä pidempi, enintään 4000-merkkinen kuvaus, jonka käyttäjät näkevät halutessaan saada lisätietoja sovelluksesta. (Google Play Developer Console Help 2016i.)

Sanallisten kuvausten lisäksi sovelluksesta on kuvion 25 mukaisesti lähetettävä vähintään kaksi kuvaa, jotka voivat olla joko kuvakaappauksia sovelluksesta tai muita sovelluksen toimintoja tai ominaisuuksia esitteleviä kuvia. Kuvien lisäksi sovellusta voidaan esitellä myös videon avulla. Myös eri laitetyppeille, kuten tableteille tai älykelloille, on sovelluksen niitä tukeessa mahdollista lisätä jokaiselle omat sovellusta esittelevät kuvat. Joista laitetyppeistä kohden voi lisätä enintään kahdeksan kuvaa. Sovelluksen toimintaa esittelevien kuvien lisäksi sovellukselle on lisättävä kuvake,

joka näkyy sovelluksen yhteydessä Google Play -kaupassa. Kuvallisille sisällöille sekä niiden resoluutioille että tiedostomuodoille on määritetty rajoitukset ja suunnitteluvaatimukset, joita on noudatettava. (Google Play Developer Console Help 2016d.)

Sovelluksen kuvauksesta ja kuvallisesta sisällöstä on mahdollista tehdä eri kielille lokalisoinnit, jotka näkyvät käyttäjien kieliasetusten mukaisesti. Kuvauksen lokalisointiin voi halutessaan käyttää Googlen tarjoamaa maksullista ammattikäänntäjän käännöspalvelua tai ilmaista automaattikäännoä. Lokalisointi ei kuitenkaan ole pakollista, jolloin sovelluksen tuotetiedot näytetään käyttäjille sovelluksen oletuskielellä. (Google Play Developer Console Help 2016i.)

GRAPHIC ASSETS

Phone Tablet Android TV Android Wear

Hi-res icon *
Default – Finnish – fi-FI
512 x 512
32-bit PNG (with alpha)

Feature Graphic *
Default – Finnish – fi-FI
1024 w x 500 h
JPG or 24-bit PNG (no alpha)

Promo Graphic
Default – Finnish – fi-FI
180 w x 120 h
JPG or 24-bit PNG (no alpha)

TV Banner
Default – Finnish – fi-FI
320 w x 180 h
JPG or 24-bit PNG (no alpha)

CATEGORIZATION

Application type * Applications

Category * Business

Content rating * Low Maturity

Some rating options have been disabled based on the content of your application. [Learn more about content rating.](#)

New content rating *

APPLIED RATING
IARC Certificate ID:
06736173-a685-466d-b460-01557cafb60
Submitted: Jul 8, 2015, 2:35 PM
[View details](#) [Learn more](#)

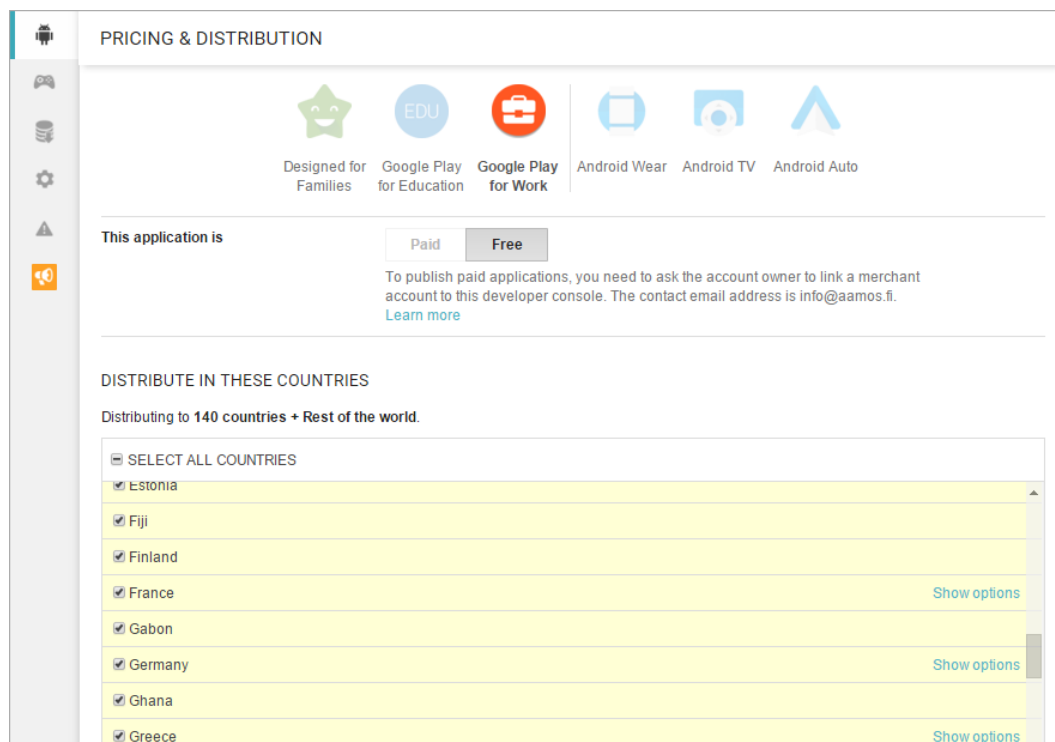
L E 3 USK 0 3+

KUVIO 25. Kuvallisen sisällön, kategorian ja ikärajan määrittäminen

Google Play -kauppaan julkaistavalle sovellukselle valitaan kuvion 25 kaltaisesti sovellustyyppi ja kategoria. Sovellukset jaetaan kaupassa kahteen eri pääkategoriaan: sovelluksiin ja peleihin. Sovelluksille on valittavissa erilaisia kategorioita, kuten viihde, liiketoiminta, valokuvaus, uutiset ja työkalut. Pelien tapauksessa mahdollisia kategorioita ovat esimerkiksi toiminta-, seikkailu-, opetus-, urheilu- ja tietopelit. (Google Play Developer Console Help 2016g.) Julkaistavilla sovelluksilla on oltava myös ikärajoitukset. Ikärajoitus määräytyy sovelluksen sisältöön liittyvän kyselyn perusteella. (Google Play Developer Console Help 2016b.)

Lopuksi sovelluksen tietoihin on määritettävä yhteystiedot, joiden avulla käyttäjät voivat ottaa kehittäjään yhteyttä muun muassa mahdollisissa ongelmatilanteissa. Yhteydenotto tapahtuu ensisijaisesti sähköpostin välityksellä, mutta muita mahdollisia yhteydenottokanavia ovat lisäksi puhelin ja verkkosivu. Viimeisenä asiana tuotetietoihin liittyen kehittäjällä on mahdollisuus lisätä sovellukselle tietosuojakäytäntö, josta selviää esimerkiksi sovelluksessa mahdollisesti pyydettyjen henkilötietojen käsittelyn tarkoitus ja niiden säilyttämiseen ja luovuttamiseen liittyvät asiat (Henkilötietolaki 523/1999, 10 §). (Google Play Developer Console Help 2016i.)

Tuotetietojen jälkeen siirrytään määrittämään sovelluksen hinnoittelua ja jakelua kuvion 26 mukaiseen näkymään. Google Play -kaupassa on mahdollista jakaa sekä ilmaisia että maksullisia sovelluksia. Maksullisia sovelluksia varten kehittäjätalille on määritettävä maksullisten sovellusten jakamiseen liittyvät maksutiedot. Maksullisen sovelluksen hintatiedot voi määrittää samanlaisiksi jokaiseen maahan, jossa sovellusta kaupataan, tai vaihtoehtoisesti hintatietoja voidaan muokata maakohtaisesti. Sekä maksullisissa että ilmaisissa sovelluksissa sovelluksen jakelu voidaan rajoittaa ainoastaan joihinkin valittuihin maihin. (Google Play Developer Console Help 2016c.)



KUVIO 26. Sovelluksen hinnoittelun ja jakelun määrittäminen kehittäjäkonsolissa

Sovelluksen hinnoittelun ja jakelun lisäksi kuvion 26 mukaisella sivulla ilmoitetaan myös, sisältääkö sovellus mainoksia. Tieto sovelluksen sisältämisestä mainoksista ilmoitetaan käyttäjälle Google Play -kaupan tietosivulla. Mainosten sisältämisen lisäksi voidaan ilmoittaa sovelluksen toimivan myös älykelloille, televisioille ja autoille kehitetyissä Androidin käyttöjärjestelmäversioissa. Mikäli sovellus on kehitetty ensisijaisesti lapsia ja perheitä, töitä tai opiskelua varten, on sovellus mahdollista lisätä jaeltavaksi kyseisten aiheiden alaisissa osioissa. Sovellusten jakeluun liittyen on myös ilmoitettava, haluaako sovellusta markkinoitavan Googlen toimesta sekä noudattaako sovellus Androidin sisältöohjeistuksia ja Yhdysvaltain viientisäännöksiä. (Google Play Developer Console Help 2016c.)

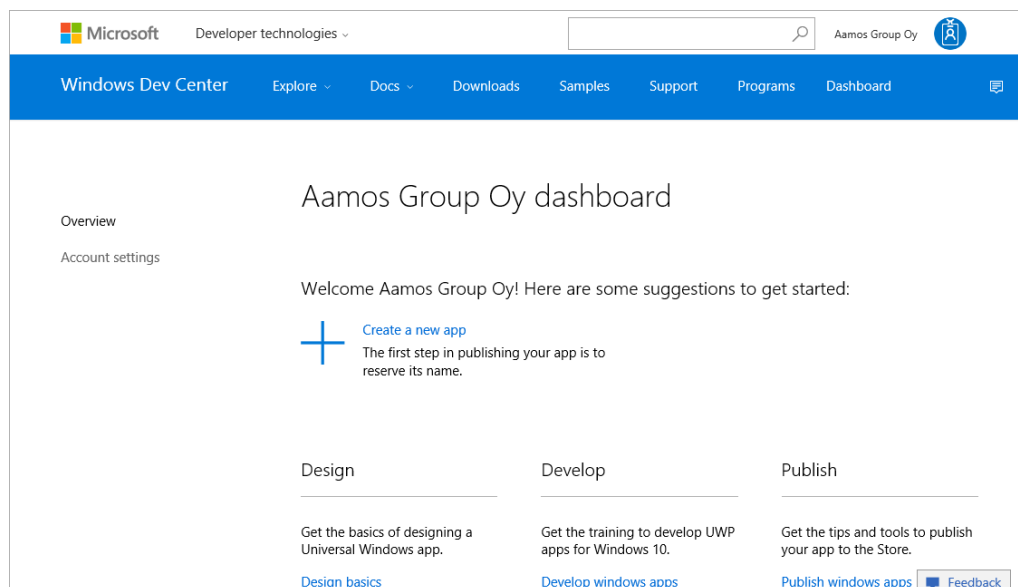
Kaikkien edellä läpikäytyjen vaiheiden jälkeen sovellus ja sen tiedot ovat valmiita julkaistaviksi Google Play -kauppaan. Sovellus julkaistaan Google Play -kaupassa sen käsittelyn jälkeen. Sovellus voidaan hylätä, jos se rikkoo Google Play -kaupan käytäntöjä, eikä sitä täten julkaista ennen kuin sovelluksesta on poistettu tai muutettu käytäntöjä rikkovat ominaisuudet. (Google Play Developer Console Help 2016f.)

4.2 Windows Phone -kauppa

4.2.1 Kehittäjätilin luonti

Sovelluksen saattamiseksi Windows Phone -kauppaan tarvitaan kehittäjätili. Windows Phone -kauppaa varten kehittäjätilejä on kahta erilaista tyyppiä: yksilö- ja yritystili. Yksilötili on tarkoitettu yksityisille kehittäjille, joilla ei ole sovelluskehityksen taustalla yritystoimintaa. Yritystili sen sijaan on tarkoitettu nimensä mukaisesti yrityksille. Yksilötili on yritystiliä edullisempi 14 euron kertamaksullaan, kun yritystilistä veloitetaan 75 euron kertamaksu. Yritystili todennetaan sen luomisen jälkeen oikean, olemassa olevan yrityksen luomaksi. Yritystili antaa enemmän mahdollisuuksia käyttää eri Windows Phone -käyttöjärjestelmän toiminnallisuuksia. Yritystilin mahdollistamia toiminnallisuuksia ovat yrityksen sisäisen lähiverkon eli intranetin käyttö, erilaisten käyttäjävarmenteiden hyödyntäminen ja pääsy käyttäjän puhelimesta sijaitseviin tiedostoihin. (Windows Dev Center 2016a.)

Kehittäjätilityypin valinnan lisäksi tiliä luotaessa on syötettävä muita kehittäjään liittyviä tietoja. Näihin tietoihin lukeutuvat kehittäjän kotimaa, sähköposti, lähiosoite sekä kehittäjänimi, joka näkyy Windows Phone -kaupassa sovelluksen kehittäjänä. Yritystilin tapauksessa kehittäjänimen on oltava sama kuin yrityksen nimi. Lisäksi yritystilille on määritettävä yhteyshenkilön nimi, sähköpostiosoite ja puhelinnumero. Tarvittavien tietojen syöttämisen jälkeen siirrytään kehittäjätilin maksamiseen, jonka suoritettuaan kehittäjällä on pääsyoikeus kuvion 27 mukaiseen Windows Phone -kaupan kehittäjäkeskukseen. Sen kautta kehittäjän on mahdollista julkaista sovelluksia Windows Phone -kaupassa. (Windows Dev Center 2016l.) Yhdelle kehittäjätilille voi lisätä useita alikäyttäjiä hallitsemaan sovelluksien julkaisemista (Windows Blog 2016).



KUVIO 27. Windows Phone -kaupan kehittäjäkeskus

4.2.2 Sovelluksen asennustiedoston luonti

Windows Phone -kauppaan julkaistavalle sovellukselle on varattava nimi, koska jokaisella kaupassa olevalla sovelluksella on oltava oma yksilöllinen nimensä. Nimen varaaminen tapahtuu Windows Phone -kaupan kehittäjäkeskuksen kautta kuvion 28 mukaisesti kirjoittamalla haluamansa nimen ja tarkastamalla, onko kyseinen nimi käytettävissä. Asennustiedoston luonti ja sovelluksen julkaiseminen kauppaan eivät onnistu ilman sovelluskohtaisen nimen varaamista. (Windows Dev Center 2016g.)

The screenshot shows a dialog box titled 'Create your app by reserving a name'. It contains the following text: 'Once you reserve a name, your app will be provisioned for services like push notifications and you can start defining IAPs (in-app products). Make sure you have the rights to use any name you reserve. You must submit this app to the Store within one year, or you'll lose your name reservation. [Learn more](#)'. Below this text is a text input field containing the name 'Aamos TAS työajanseuranta', followed by a green checkmark icon. At the bottom of the dialog are two buttons: 'Reserve app name' (in blue) and 'Cancel' (in grey).

KUVIO 28. Nimen varaaminen sovellukselle

Sovelluksen nimen varaamisen jälkeen voidaan aloittaa itse asennuspaketin luominen, joka tapahtuu Visual Studio -sovelluskehitysympäristön Create App Packages -työkalun avulla. Asennuspakettia luotaessa valitaan aiemmin varattu sovelluksen nimi, jonka työkalu automaattisesti liittää sovelluksen tietoihin yhdessä muiden asennuspaketin luomista varten tarvittavien tietojen, kuten sovelluksen julkaisijan eli kehittäjän, kanssa. Tietojen liittämisen jälkeen siirrytään kuvion 29 mukaisesti määrittämään tarkemmin asennuspaketin tietoja ja luotavia asennuspaketteja. (Microsoft Virtual Academy 2014e.)

Create App Packages

Select and Configure Packages

Output location:
D:\Temp\Aamos TAS työajanseuranta\AppPackages\

Version:
1 . 1 . 0 . 0
☒ Automatically increment

Generate app bundle:
If needed
[What does an app bundle mean?](#)

Select the packages to create and the solution configuration mappings:

Architecture	Solution Configuration
<input checked="" type="checkbox"/> Neutral	Release (Any CPU)
<input type="checkbox"/> x86	Release (x86)
<input type="checkbox"/> ARM	Release (ARM)

i The option to create x86 package is disabled because it's not a valid package that can be published to the Windows Phone Store.

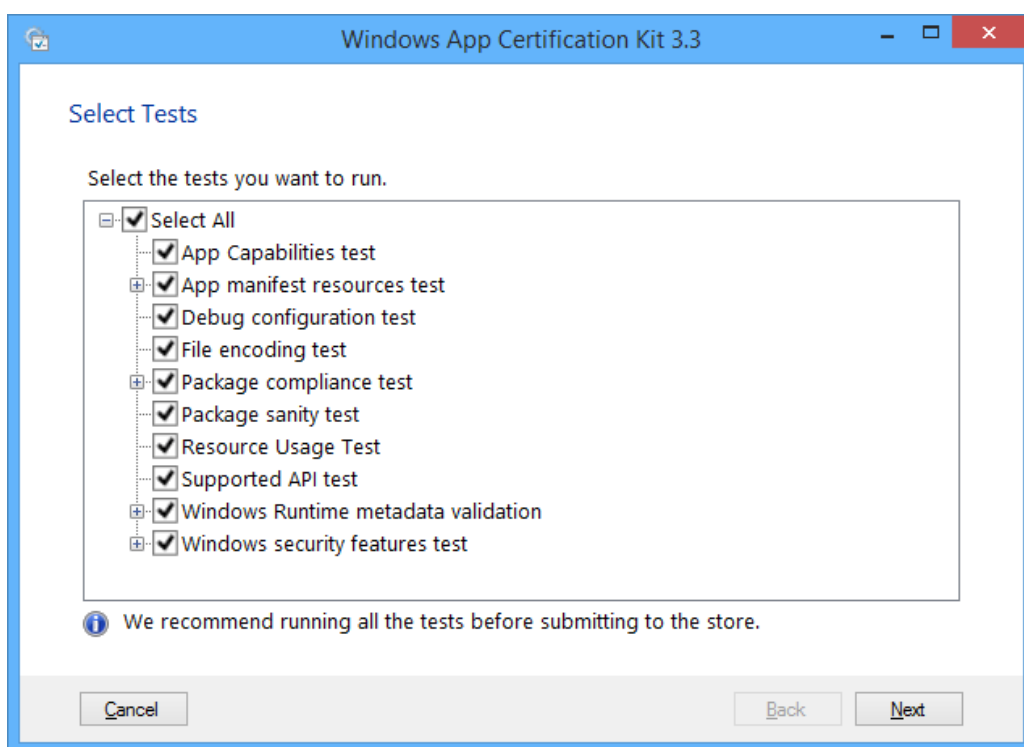
☒ Include public symbol files, if any, to enable crash analysis for the app

Previous Create Cancel

KUVIO 29. Asennuspaketin tietojen määrittäminen

Ensin kuvion 29 näkymässä määritetään sijainti, jonne asennustiedosto halutaan luoda, sekä sovelluksen versionumero. Tämän jälkeen määritetään, halutaanko asennustiedostosta luoda vain yksi yhtenäinen paketti kaikkine eri versioiden sisältöineen. Sovelluksen koko kasvaa, jos sovelluksessa on paljon esimerkiksi eri kieliversioille suunnattuja sisältöjä, kuten

kuvia ja muita paljon isoja tilaa vieviä sisältöjä. Näissä tapauksissa sovelluksesta olisi hyvä luoda oma asennustiedosto jokaiselle eri kieliversiolle, jolloin asennustiedoston koko pysyisi mahdollisimman pienenä. Lisäksi on valittava koontiversio ja arkkitehtuuri, jolle asennustiedostot luodaan. Jos sovelluksessa ei ole alustariippuvaisia toteutuksia, arkkitehtuuriksi voidaan valita Neutral-vaihtoehto, joka tarkoittaa, että sovellusta voidaan käyttää kaikilla Windows Phone -mobiililaitteilla. Koontiversioksi valitaan julkaisuversio. (SubramanyamRaju Windows Phone Tutorials 2014.) Lopuksi voidaan valita, halutaanko sovelluksen epänormaalia käyttäytymisestä kerätä tietoja, joita voidaan analysoida kehittäjäkeskuksen kautta. Näiden vaiheiden jälkeen asennustiedosto on valmiina luotavaksi. (Microsoft Developer Network 2016b.)



KUVIO 30. Sovelluksen testaustyökalu Windows App Certification Kit

Ennen sovelluksen julkaisemista kauppaan olisi asennustiedosto hyvä tarkistaa Windows App Certification Kit -työkalun avulla. Työkalu suorittaa sovellukselle useita kuviossa 30 lueteltuja testejä liittyen esimerkiksi sovel-

luksen suorituskykyyn ja rajapintojen käyttöön sekä varmistaa, että sovel-
lus läpäisee testit ja on täten valmis julkaistavaksi Windows Phone -kau-
passa. (Microsoft Virtual Academy 2014e.)

4.2.3 Sovelluksen julkaiseminen kauppaan

Sovelluksen julkaiseminen Windows Phone -kauppaan aloitetaan kehittä-
jäkeskuksessa kuvion 31 mukaisessa näkymässä. Tästä näkymästä näh-
dään yleiskatsaus kaikista tiedoista, joita tarvitaan sovelluksen julkaise-
mista varten. Tarvittavat tiedot ovat hinnoittelu ja jakelu, sovelluksen omi-
naisuudet, ikäluokitukset, asennustiedostot ja sovelluksen kuvaus sekä
mahdolliset lisätiedot liittyen sovelluksen tarkistamiseen. Tämän näkymän
kautta siirrytään määrittämään kyseisiä tietoja yksityiskohtaisemmin. Ky-
seiset tiedot voi täyttää haluamassaan järjestyksessä, mutta seuraavassa
tiedot täytetään kuvion 31 mukaisessa järjestyksessä. (Windows Dev Cen-
ter 2016d.)

The screenshot displays the Windows Dev Center submission interface with the following sections and status indicators:

- Pricing and availability:** Complete (green checkmark). Subtext: Free and available to customers.
- App properties:** Complete (green checkmark). Subtext: Business.
- Age ratings:** Complete (green checkmark). Displays various age rating icons (E, 3, 3+, L, 3+, A, O, 0+, 3).
- Packages:** Complete (green checkmark). Subtext: Tyoajanseuranta_1.1.0.0_AnyCPU.ap... Validated.
- Descriptions:** Complete (green checkmark). Subtext: Finnish (Finland) Complete.
- Notes for certification:** Optional (radio button).

A blue button labeled "Submit to the Store" is located at the bottom left of the interface.

KUVIO 31. Sovelluksen julkaisemisen päänäkökulma

Ensimmäisenä siirrytään määrittämään sovelluksen hinnoittelua ja saatavuutta kuvion 32 tapaisesti. Ensin on valittava, haluaako sovellusta jaettavan ilmaiseksi vai onko se maksullinen. Maksullisten sovellusten yhteydessä julkaisijalla on mahdollisuus asettaa sovellukselle ilmainen kokeilu-aika, joka voi kestää enimmillään 30 päivää ja jonka aikana käyttäjä voi päättää, onko sovellus hyödyllinen ja haluaako hän maksaa siitä. Kokeiluajan lisäksi sovellus voidaan asettaa tietyksi ajaksi alennusmyyntiin. Sovellusta julkaistaessa on myös määritettävä, missä maissa sovelluksen halutaan olevan saatavilla ja poikkeako sovelluksen hinta kyseisessä maassa aiemmin asetetusta perushinnasta. (Windows Dev Center 2016d.)

Pricing and availability

Base price*

Free

Free trial

No free trial

Distribution and visibility

[Hide options](#)

☒ Make this app available in the Store

☐ Hide this app in the Store. Customers with a direct link to the app's listing can still download it, except on Windows 8 and Windows 8.1. [Learn more](#)

☐ Hide this app and make it available only to the people you specify below, who can download this app on Windows Phone 8.x devices. A promotional code may be used to download this app on Windows 10 devices. [Learn more](#)

☐ Hide this app and prevent acquisition. Customers with a promotional code can still download it on Windows 10 devices. [Learn more](#)

Publish date

☒ Publish this app as soon as it passes certification.

☐ Publish this app manually.

☐ No sooner than

02/17/2016

14:00

UTC

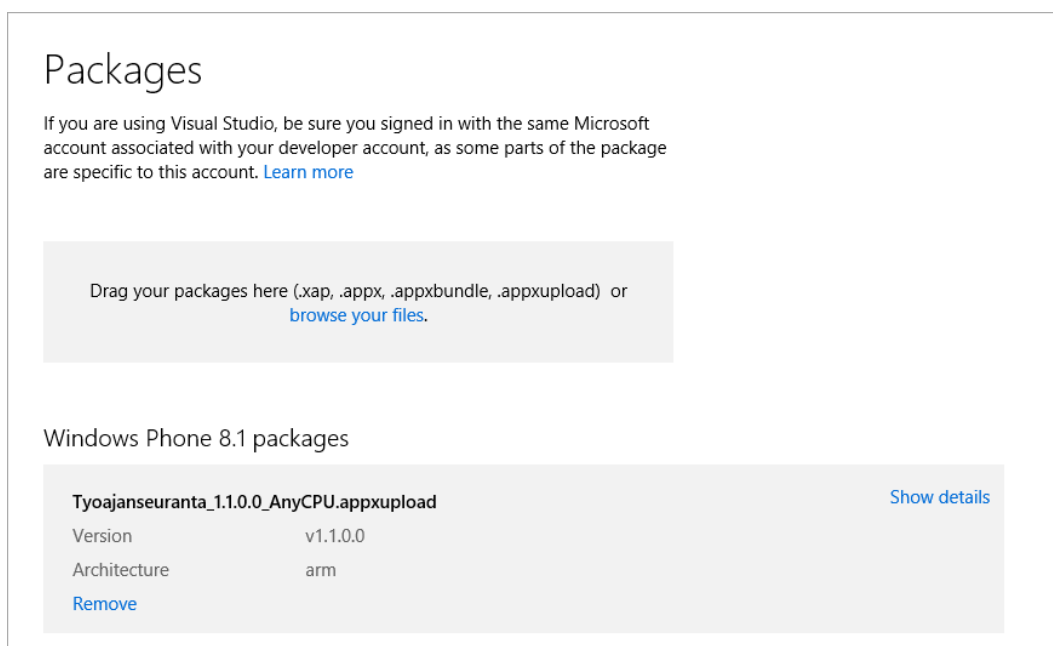
KUVIO 32. Sovelluksen hinnoittelun ja jakelun määrittäminen

Sovelluksen jakeluun ja näkyvyyteen liittyen on mahdollista tehdä muutamia rajoituksia. Sovellus voidaan asettaa joko kaikkien nähtäville tai piilottaa se vain joidenkin käyttäjien nähtäväksi. Kaupasta piilotettua sovellusta voidaan jakaa käyttäjille sovelluskohtaisen linkin tai kampanjakoodin avulla tai vaihtoehtoisesti määrittämällä niiden käyttäjien Microsoft-tilin sähköpostiosoitteet, joille sovellus halutaan jakaa. Lopuksi valitaan, milloin sovellus halutaan julkaista Windows Phone -kaupassa. Julkaisuajankohdaksi voidaan valita automaattinen julkaisu heti, kun sovellus on läpäissyt tarkistuksen, manuaalinen julkaisu, jolloin julkaisija julkaisee sovelluksen haluamanaan ajankohtana sen läpäistyä tarkistuksen, tai määrittämällä ajankohta, jolloin sovellus halutaan aikaisintaan julkaista. (Windows Dev Center 2016n.)

Hinnoittelun ja jakelun määrittämisen jälkeen siirrytään asettamaan sovelluksen ominaisuuksia. Sovellukselle on valittava sille sopiva kategoria ennalta määritetyistä vaihtoehtoista. Valittavia kategorioita ovat esimerkiksi uutiset, yhteisöpalvelut, liiketoiminta, työkalut, koulutus ja pelit. Joillakin kategorioilla on lisäksi alikategorioita. Käyttäjille on mahdollista ilmoittaa kategorian lisäksi myös sovelluksen laitteistomääriykset, jotta he tietävät, toimiiko sovellus heidän laitteessaan, kuten on tarkoitettu. Valittavana on muutamia esimääritettyjä ominaisuuksia, kuten kamera, NFC-lähiluku ja kosketusnäyttö. Käyttäjä saa ennen sovelluksen lataamista ilmoituksen, jos hänen laitteensa ei täytä kaikkia laitteistomääriyksiä. Laitteistomääritysten lisäksi julkaisija voi ilmoittaa käyttäjille tietoa myös muista määriyksistä. Tällaisia ovat erilaisten hankintojen tekeminen sovelluksessa Windows Phone -kaupan ulkopuolelta, helppokäyttöisyysohjeiden noudattaminen, mahdollisuus asentaa sovellus muistikortille ja sovelluksen varmuuskopioiden automaattinen tallentaminen käyttäjän Microsoft-tilin OneDrive-pilvipalveluun. (Windows Dev Center 2016h.)

Windows Phone -kaupassa julkaistavilla sovelluksilla on oltava ikäluokitus. Ikäluokitus määritetään Windows Phone -kaupan ikäluokituksista vastaavan kansainvälisen ikäluokitusyhdistyksen IARC:n (International Age Ratings Coalition) teettämän kyselyn perusteella. Kyselyssä esitettävät kysy-

mykset riippuvat sovelluksen kategoriasta ja ne liittyvät sovelluksen sisältöön. Kyselyssä kysytään tietoa esimerkiksi siitä, sisältääkö sovellus loukkaavaa kielenkäyttöä, huumeviittauksia tai joko väkivaltaista tai seksuaalista materiaalia sekä onko sovelluksen avulla mahdollista olla vuorovaikutuksessa toisten käyttäjien kanssa. Kyselyyn vastaamisen jälkeen sovelluksen ikäluokitus määräytyy automaattisesti vastausten perusteella. (Windows Dev Center 2016b.)



KUVIO 33. Asennuspaketin lisääminen

Ikäluokituksen määrittämisen jälkeen Windows Phone -kaupan kehittäjäkeskukseen ladataan kuvion 33 mukaisen näkymän kautta aiemmin luotu asennuspaketti. Samasta sovelluksesta on mahdollista ladata useampi asennuspaketti kaikille mahdollisille sovelluksen tukemille käyttöjärjestelmille. Asennuspaketin lataamisen yhteydessä paketti validoidaan erilaisten virheiden varalta. (Windows Dev Center 2016d.) Mahdolliset virheet voivat liittyä esimerkiksi asennuspaketin tyyppiin tai versionumerointiin (Windows Dev Center 2016m).

Lopuksi sovellukselle lisätään kuvaus kuvion 34 mukaisella sivulla. Sovelluksen kuvaus koostuu sekä sanallisesta että kuvallisesta informaatiosta.

Sovelluksen kuvaus voidaan lisätä joko yhdellä tai jokaisella sovelluksen vaatimusmäärittelytiedostossa asetetulla kielellä. Myös eri käyttöjärjestelmille voidaan luoda omat kuvaukset. Pakollisina tietoina kuvaukseen on syötettävä enimmillään 10 000 merkkiä pitkä kuvausteksti ja vähintään yksi sovelluksen näyttökuva. Näiden lisäksi sovelluksen kuvaukseen on mahdollista lisätä monia muita erilaisia tietoja. (Windows Dev Center 2016d.)

App description - Finnish (Finland)




Default description Hide

Description*


Aamos TAS on kalenteripohjainen ja helpokäyttöinen työajanseurantaan kehitetty ohjelma. Huom! Ohjelma tarvitsee toimiakseen kalenteriin lisätyt merkintäpohjat, jotka saat Aamos Group Oy:stä, puh. 0400 972 165 tai sähköposti office365@aamos.fi.


- Character limit: 10000
- [Learn how to write a good description](#)

Mobile screenshots

- 768 x 1280, 720 x 1280, or 480 x 800 pixels
- Max upload 8 files
- Landscape or portrait
- Accepted file types: .png
- Less than 2 MB

Add image caption 

Add image caption 

KUVIO 34. Sovelluksen kuvauksen lisääminen

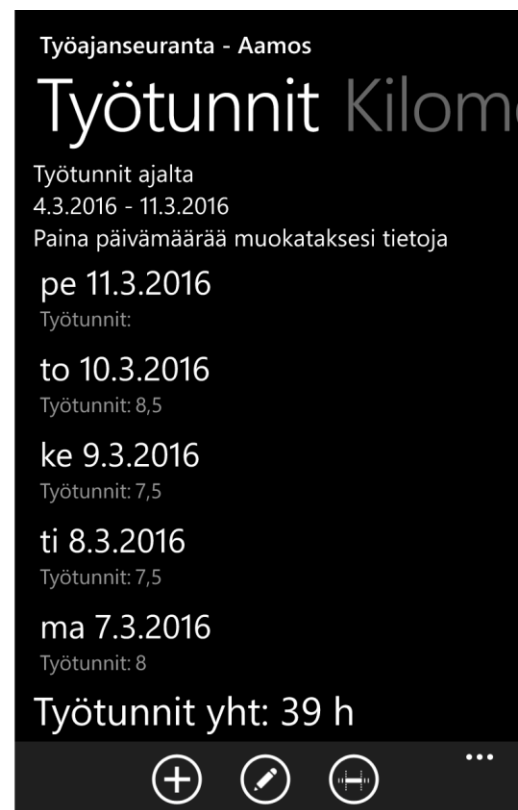
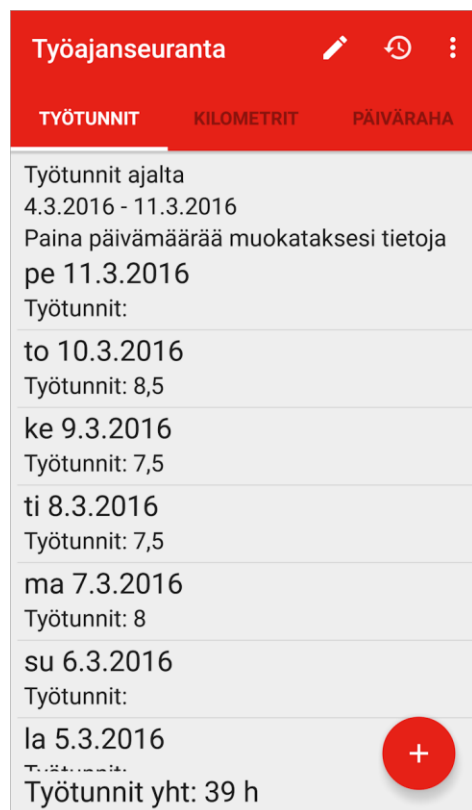
Vaihtoehtoihin tietoihin kuuluu esimerkiksi sovelluksen päivitystä julkaistessa kuvaus, jolla voidaan ilmoittaa päivityksen tuomat uudet ominaisuudet tai tehdyt muutokset enimmillään 1 500 merkillä. Kuvallista sisältöä voidaan lisätä erilaisten markkinointikuvien muodossa, joita käytetään sovelluksen markkinointiin Windows Phone -kaupassa. Lisäksi Windows Phone 8.1:lle tai sitä aikaisemmille mobiilikäyttöjärjestelmäversioille on liisättävä sovelluksen kuvake tai muutoin sovelluksella ei ole Windows Phone -kaupassa kuvaketta. Sovelluksen ominaisuuksia ja suositeltuja laitteita voidaan esitellä tarkemmin enintään 200 merkin pituisilla kuvauksilla. Ominaisuuskohtaisia kuvauksia on mahdollista luoda yhteensä 20 ja

laitekohtaisia 11. Sovelluksen esiintymistä Windows Phone -kaupan hakutuloksissa voidaan parantaa lisäämällä sovellukselle enintään kahdeksan sovellusta kuvaavaa avainsanaa. Sovellukselle voidaan määrittää tarvittaessa erilaisia tekijänoikeus- ja tavaramerkkitietoja, käyttöoikeusehtoja sekä tietosuojakäytäntöjä. Julkaisija voi myös lisätä sovelluksen tietoihin yhteydenottokanavia verkkosivuston ja sähköpostiosoitteen muodossa. (Windows Dev Center 2016d.)

Kaikkien pakollisten tietojen asettamisen jälkeen sovellus on valmis lähettäväksi tarkistukseen kuvion 31 mukaisen sovelluksen julkaisun päänäköymän kautta. Ennen sovelluksen lähettämistä on kuitenkin annettava mahdolliset tiedot, joita tarvitaan sovelluksen tarkistamiseksi. Tällaisia tietoja voivat olla esimerkiksi palveluun tarvittavat käyttäjätunnukset sekä ohjeet piilotettujen tai lukittujen sisältöjen avaamiseen (Windows Dev Center 2016k). Tarkistukset liittyvät sovelluksen turvallisuuteen ja sisällön ohjeidenmukaisuuteen. Mikäli sovellus ei läpäise tarkistuksia, julkaisija saa raportin hylkäykseen johtaneista syistä. Sovelluksen läpäistyä sille suoritettut tarkistukset on sovellus valmis julkaistavaksi Windows Phone -kaupassa julkaisijan määrittämänä ajankohtana. (Windows Dev Center 2016p.)

5 SOVELLUKSEN TOTEUTUS

Sovellus avautuu käynnistyessään joko kuviossa 35 esiintyvään oikean- tai vasemmanpuoleiseen näkymään käyttöjärjestelmästä riippuen. Eri työmerkinnät, kuten työtunnit, kilometrit ja päivärahat, ovat tarkasteltavissa omilla välilehdillään. Oletuksena työmerkintöjä näytetään viimeisen viikon ajalta. Jokaisella välilehdellä on myös yhteenlaskettu muun muassa työtuntien ja kilometrien yhteismäärä valitulta aikaväliltä. Aloitusnäköymästä voidaan siirtyä kaikkiin sovelluksen toimintoihin, joita ovat kaikkien päivän työmerkintöjen kirjaaminen yhdessä näkymässä, yksittäisen työmerkinnän kirjaaminen ja tarkastelu sekä uuden tapahtuman tai työtehtävän lisääminen kalenteriin. Sovelluksen ulkoasussa on pyritty noudattamaan kunkin käyttöjärjestelmän omaa yleistä ulkoasua.



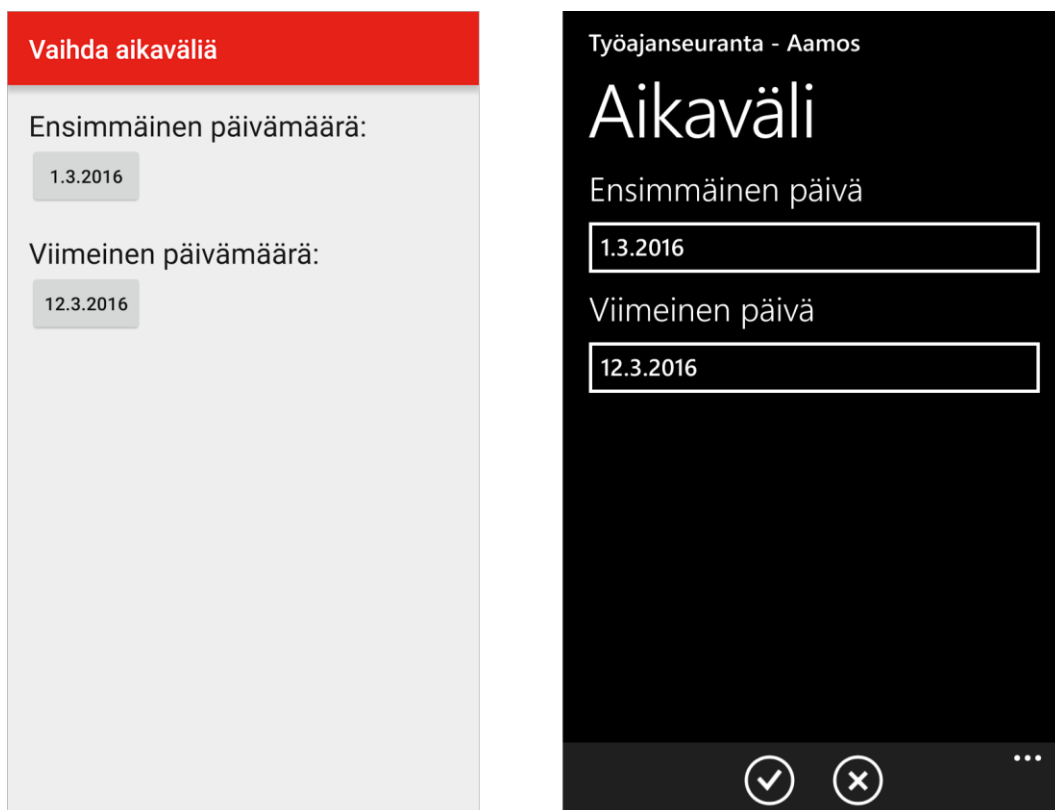
KUVIO 35. Sovelluksen aloitusnäkömät Android- (kuviossa vasemmalla) ja Windows Phone -mobiilikäyttöjärjestelmillä

Ensimmäistä kertaa sovellusta käynnistettäessä käyttäjän on määritettävä asetukset kuvion 36 mukaisessa näkymässä. Määritettävänä ovat yritys ja kalenteri. Eri yritykset saattavat haluta kerätä erilaisia työmerkintöjä työntekijöiltä, jolloin yritys-valinta vaikuttaa siihen, mitä eri työmerkintöjä sovelluksen kautta voidaan kirjata. Esimerkiksi sovelluksen päänäkylässä olevat työmerkintä-välilehdet määräytyvät tämän asetuksen mukaan. Kalenteri-valintaan käyttäjä valitsee oman työpaikkansa kalenterin, jossa työmerkinnät ja -tehtävät sijaitsevat. Valittavana olevat kalenterit haetaan luvussa kolme kerrotulla tavalla. Asetukset tallennetaan puhelimen muistiin, ja niitä on mahdollista muuttaa jälkikäteen.

KUVIO 36. Asetusten määrittäminen

Asetusten määrittämisen jälkeen voidaan hakea aloitusnäkylässä näytettävät työmerkinnät. Työmerkintöjen haku kalenterista suoritetaan luvussa kolme esitetyn kalenterimerkintöjen hakemisen mukaisesti. Aikaväliä, jolta työmerkinnät haetaan, voidaan muuttaa kuvion 37 mukaisen näkymän

kautta, määrittämällä välin ensimmäinen ja viimeinen päivä. Windows Phone -käyttöjärjestelmä rajoittaa menneiden kalenterimerkintöjen hakemisen enimmillään kahteen edelliseen viikkoon (Windows Phone How-to 2016). Valittu aikaväli palautuu oletukseksi sovelluksen seuraavalla käynnistyskerralla, jotta käyttäjän ei tarvitsisi itse vaihtaa aikaväliä nähdäkseen ajankohtaisimmat työmerkinnät.



KUVIO 37. Aikavälin vaihtaminen

Aloituskäytöstä voidaan siirtyä kirjaamaan tai tarkastelemaan listassa olevia työmerkintöjä valitsemalla haluttu merkintä. Mikäli työmerkintää ei ole vielä kirjattu, siirrytään kuvion 38 vasemmanpuoleiseen näkymään, jossa kyseinen tieto, esimerkiksi työtunnit, voidaan syöttää. Normaalisti syötettävänä on vain yksi tieto, mutta kilometrejä kirjattaessa kirjattavia tietoja on useampia, jolloin näkymä on kuvion 38 keskimmäisen näkymän mukainen. Kilometrejä kirjattaessa syötetään kuljettu matkareitti, auton mittarilukema matkan alussa ja lopussa sekä matkan aikana omaksi ajoksi

laskettavat kilometrit, joista lasketaan automaattisesti korvattavat kilometrit. Edellisen päivän mittarin loppulukema voidaan hakea ja lisätä aloituslukeksi automaattisesti. Työmerkintään kirjatut tiedot tallennetaan kalenteriin luvussa kolme esitellyn kalenterimerkintöjen päivittämisen mukaisesti. Tietojen kirjaamisen ja tallentamisen jälkeen siirytään aloitusnäky-
mään. Windows Phone -käyttöjärjestelmällä kalenterimerkintöihin tehtävät muutokset pitää käydä tallentamassa järjestelmän oman kalenterisovelluksen kautta (Microsoft Virtual Academy 2014d). Valittaessa listasta työmerkintä, jonka tiedot on jo merkitty, siirytään kuvion 38 oikeanpuolimmais-
seen näkymään. Näkymässä nähdään yleiskatsaus valitusta työmerkinnästä ja voidaan siirtyä tarvittaessa muokkaamaan merkintää.

KUVIO 38. Työmerkinnän kirjaaminen ja tarkastelu

Työmerkintöjen kirjaaminen on mahdollista tehdä kuluvan päivän osalta myös yhdellä kertaa. Tällöin kaikki kirjattavat työmerkinnät näkyvät samassa, kuvion 39 mukaisessa näkymässä. Näkymään haetaan kaikki kuluvan päivän työmerkinnät, jotka ovat käyttäjän kirjattavissa. Käyttäjä kirjaa haluamansa tiedot, minkä jälkeen ne tallennetaan ja sovellus palaa takaisin aloitusnäky-
mään.

The image displays two versions of a mobile application interface for logging work data. The left version is light-themed with a red header bar containing the text 'Kirjaa tiedot' and a clipboard icon. Below the header, the date 'la 12.3.2016' is shown. The form includes input fields for 'Työtunnit', 'Kilometrit', 'Matkareitti', 'Mittarilukema alussa', 'Mittarilukema lopussa', and 'Omaa ajoa'. A red circular button with a white checkmark is positioned at the bottom right. The right version is dark-themed with a black header bar containing 'Työajanseuranta - Aamos' and 'Kirjaa tiedot'. The date '12.3.2016' is displayed below the header. The same form fields are present, but the 'Omaa ajoa' field is partially obscured. A bottom navigation bar features three icons: a clipboard, a right arrow, and a menu (three dots).

KUVIO 39. Kaikkien työmerkintöjen kirjaaminen yhdessä näkymässä

Uusien kalenterimerkintöjen ja työtehtävien lisäämisessä kalenteriin hyödynnetään puhelimen omaa oletuskalenterisovellusta. Uutta merkintää liittäessä avataan kalenterisovellus, jonka kautta täytetään merkintään liittyvät tiedot, kuten aihe, tapahtuman kesto, sijainti ja muut merkintään liittyvät lisätiedot. Ennen siirtymistä kalenterisovellukseen, kuvion 40 vasemmanpuoleisen mukaisen näkymän avulla voidaan tehdä kuitenkin muutamia lisävalintoja. Käyttäjä voi valita, haluaako hän lisätä luotavan merkinnän keston automaattisesti kyseisen päivän työtunteihin. Sovellus ottaa huomioon mahdolliset päällekkäisyydet muiden tapahtumien kanssa. Käyttäjän on myös mahdollista paikantaa nykyinen sijaintinsa ja lisätä paikkatieto kalenterimerkinnän lisätieto-kenttään. Käyttäjän lisäämät merkinnät näkyvät valitulta aikaväliltä kuvion 40 oikeanpuoleisen näkymän mukaisesti omassa näkymässään, jonka kautta niiden tietoja voidaan tarkastella ja muokata. Merkintöjen muokkaamista varten siirrytään uuden merkinnän luomisen tavoin puhelimen oletuskalenterisovellukseen.

Uusi merkintä


☒ Lisää tapahtuman kesto päivän työtunteihin



☒ Lisää nykyinen sijaintisi merkinnän lisätietoja-kenttään

HAE SIJAINTI

PERUUTA

Koordinaatit:
Leveysasteet: 60.670093°
Pituusasteet: 25.854681°
Osoite: Keskustie 14
07600 Myrskylä



Tapahtumat  

1.6.2015 - 30.6.2015
Paina tapahtumaa nähdäksesi lisätietoja
Työajanseuranta mobiilisovelluks...
ti 16.6.2015
08:00


Työajanseuranta mobiilisovelluks...
ma 15.6.2015
08:00

Työajanseuranta mobiilisovelluks...
pe 12.6.2015
08:00

Työajanseuranta mobiilisovelluks...
to 11.6.2015
08:00

Asiakkaan tietokoneen päivitys
ke 10.6.2015
14:00

Työajanseuranta mobiilisovelluks...
ke 10.6.2015
08:00



KUVIO 40. Uuden merkinnän lisääminen ja merkintöjen tarkastelu

6 YHTEENVETO

Työssä toteutettiin Aamos Group Oy:n kehittämää kalenteripohjaista työajanseuranta-palvelua varten natiivi mobiilisovellus Android- ja Windows Phone -mobiilikäyttöjärjestelmille. Toteutetun mobiilisovelluksen tarkoituksena on helpottaa erilaisten työajanseurantaan liittyvien työmerkintöjen, kuten työtuntien, kilometrien ja päivärahojen, kirjaamista palveluun älypuhelimella. Kehitetty mobiilisovellus julkaistiin molempien mobiilikäyttöjärjestelmien omassa sovelluskaupassa.

Mobiilisovellukseen haluttiin erilaisia käyttöä helpottavia ominaisuuksia, joita olivat työmerkintöjen kirjaaminen joko yksittäin tai yhdessä saman näkymän kautta, uusien työtehtävien luominen ja niiden keston lisääminen automaattisesti päivän työtunteihin sekä paikkatiedon lisääminen työtehtävään. Kyseiset ominaisuudet toteutettiin onnistuneesti sekä Android- että Windows Phone -mobiilikäyttöjärjestelmille. Molempien sovellusten ollessa ominaisuuksiensa osalta täysin identtiset löytyy niiden ainoa ero käyttöliittymästä. Käyttöliittymästä on pyritty luomaan kunkin käyttöjärjestelmän omaa käyttöliittymää noudatteleva käyttökokemuksen helppouden ja tuttavallisuuden takaamiseksi.

Työajanseuranta-mobiilisovelluksen kehittämisen suurimmat haasteet liittyivät enimmäkseen sen dynaamisuuteen. Sovelluksen päänäkyvässä olevat välilehdet on luotava dynaamisesti, koska niiden määrä ja sisältö voivat vaihdella asetuksissa valitun yrityksen mukaan. Samasta syystä johtuen myös näkymä, jossa kaikki työmerkinnät kirjataan yhdessä, luodaan dynaamisesti siinä vaihtelevien merkintöjen vuoksi.

Työajanseuranta-mobiilisovellus on kaikkien Aamos Group Oy:n työajanseuranta-palvelua käyttävien yritysten työntekijöiden käytettävissä. Käyttäjiltä saadun palautteen perusteella sovellus on toiminut moitteetta ilman minkäänlaisia virheitä. Tulevaisuudessa sovelluksesta olisi mahdollista kehittää oma versio Applen iPhone-puhelimien iOS-mobiilikäyttöjärjestelmälle sekä parantaa entisestään sovelluksen käytettävyyttä pienillä lisäominaisuuksilla.

LÄHTEET

Android Developers. 2016a. Activities [viitattu 1.3.2016]. Saatavissa: <http://developer.android.com/guide/components/activities.html>

Android Developers. 2016b. Android Studio Overview [viitattu 29.2.2016]. Saatavissa: <http://developer.android.com/tools/studio/index.html>

Android Developers. 2016c. App Manifest [viitattu 3.3.2016]. Saatavissa: <http://developer.android.com/guide/topics/manifest/manifest-intro.html>

Android Developers. 2016d. Calendar Provider [viitattu 26.1.2016]. Saatavissa: <http://developer.android.com/guide/topics/providers/calendar-provider.html>

Android Developers. 2016e. Content Providers [viitattu 26.1.2016]. Saatavissa: <http://developer.android.com/guide/topics/providers/content-providers.html>

Android Developers. 2016f. ContentResolver [viitattu 26.1.2016]. Saatavissa: <http://developer.android.com/reference/android/content/ContentResolver.html>

Android Developers. 2016g. ContentValues [viitattu 26.1.2016]. Saatavissa: <http://developer.android.com/reference/android/content/ContentValues.html>

Android Developers. 2016h. Creating an Android Project [viitattu 2.3.2016]. Saatavissa: <http://developer.android.com/training/basics/firstapp/creating-project.html>

Android Developers. 2016i. Dashboards [viitattu 2.3.2016]. Saatavissa: <http://developer.android.com/about/dashboards/index.html>

Android Developers. 2016j. Layouts [viitattu 2.3.2016]. Saatavissa: <http://developer.android.com/guide/topics/ui/declaring-layout.html>

Android Developers. 2016k. Signing Your Applications [viitattu 28.1.2016]. Saatavissa: <http://developer.android.com/tools/publishing/app-signing.html>

Google Play Developer Console Help. 2016a. Add developer account users & manage permissions [viitattu 28.1.2016]. Saatavissa: https://support.google.com/googleplay/android-developer/answer/2528691?hl=en&ref_topic=3450769

Google Play Developer Console Help. 2016b. Content ratings for apps & games [viitattu 9.2.2016]. Saatavissa: <https://support.google.com/googleplay/android-developer/answer/188189?hl=en>

Google Play Developer Console Help. 2016c. Distribute your app [viitattu 9.2.2016]. Saatavissa: https://support.google.com/googleplay/android-developer/answer/6334373?hl=en&ref_topic=3450986

Google Play Developer Console Help. 2016d. Graphic assets, screenshots, & video [viitattu 28.1.2016]. Saatavissa: <https://support.google.com/googleplay/android-developer/answer/1078870?hl=en>

Google Play Developer Console Help. 2016e. How to use the Google Play Developer Console [viitattu 28.1.2016]. Saatavissa: <https://support.google.com/googleplay/android-developer/answer/6112435?hl=en>

Google Play Developer Console Help. 2016f. Publish an app [viitattu 9.2.2016]. Saatavissa: <https://support.google.com/googleplay/android-developer/answer/6334282?hl=en>

Google Play Developer Console Help. 2016g. Select a category for your app or game [viitattu 9.2.2016]. Saatavissa: <https://support.google.com/googleplay/android-developer/answer/113475?hl=en>

Google Play Developer Console Help. 2016h. Set up alpha/beta tests [viitattu 28.1.2016]. Saatavissa: <https://support.google.com/googleplay/android-developer/answer/3131213?hl=en>

Google Play Developer Console Help. 2016i. Upload an app [viitattu 28.1.2016]. Saatavissa: https://support.google.com/googleplay/android-developer/answer/113469?hl=en&ref_topic=3450986

Henkilötietolaki 523/1999.

Microsoft Developer Network. 2016a. Configure a Windows 8.1 app package by using the manifest designer [viitattu 3.3.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/br230259.aspx>

Microsoft Developer Network. 2016b. Create a Windows 8.1 app package [viitattu 11.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/hh975357.aspx>

Microsoft Virtual Academy. 2014a. Building Apps for Windows Phone 8.1: (01) Introducing the Windows Phone 8.1 App Development Platform [viitattu 2.3.2016]. Saatavissa: <https://channel9.msdn.com/series/Building-Apps-for-Windows-Phone-8-1/01>

Microsoft Virtual Academy. 2014b. Building Apps for Windows Phone 8.1: (02) Getting Started Building Windows Runtime Apps [viitattu 2.3.2016]. Saatavissa: <https://channel9.msdn.com/series/Building-Apps-for-Windows-Phone-8-1/02>

Microsoft Virtual Academy. 2014c. Building Apps for Windows Phone 8.1: (03) Page Navigation and Data Binding in Windows Runtime Apps [viitattu 2.3.2016]. Saatavissa <https://channel9.msdn.com/series/Building-Apps-for-Windows-Phone-8-1/03>

Microsoft Virtual Academy. 2014d. Building Apps for Windows Phone 8.1: (16) Interacting with User Data: Contacts and Appointments [viitattu 27.1.2016]. Saatavissa: <https://channel9.msdn.com/Series/Building-Apps-for-Windows-Phone-8-1/16>

Microsoft Virtual Academy. 2014e. Building Apps for Windows Phone 8.1: (21) App Packaging, Monetization, and Publication [viitattu 25.2.2015].

Saatavissa: <https://channel9.msdn.com/series/Building-Apps-for-Windows-Phone-8-1/21>

Microsoft Virtual Academy. 2014f. Part 8 - Working with the package.appx-manifest [viitattu 3.3.2015]. Saatavissa: <https://channel9.msdn.com/Series/Windows-Phone-8-1-Development-for-Absolute-Beginners/Part-8-Working-with-the-package-appxmanifest>

SubramanyamRaju Windows Phone Tutorials. 2014. WindowsPhone 8.1 App Deployment: How to deploy .appx file with Application Development Tool [viitattu 11.2.2016]. Saatavissa: <http://bsubramanyamraju.blogspot.fi/2014/04/windowsphone-81-how-to-deploy-appx-file.html>

Suomen Pankki. 2016. Valuuttakurssit – Yhdysvaltain dollari (USD) [viitattu 28.1.2016]. Saatavissa: [http://www.suomenpankki.fi/fi/tilastot/valuuttakurssit/valuuttakurssit_\(ekp\)/pages/eurofxref-graph-usd.aspx](http://www.suomenpankki.fi/fi/tilastot/valuuttakurssit/valuuttakurssit_(ekp)/pages/eurofxref-graph-usd.aspx)

Työaikalaki 605/1996.

Visual Studio. 2016. Compare Visual Studio 2015 Offerings [viitattu 1.3.2016]. Saatavissa: <https://www.visualstudio.com/en-us/products/compare-visual-studio-2015-products-vs>

Windows Blog. 2016. Using Multi-user and roles in Dev Center [viitattu 17.2.2016]. Saatavissa: <https://blogs.windows.com/buildinggapps/2015/12/22/using-multi-user-and-roles-in-dev-center/>

Windows Dev Center. 2016a. Account types, locations, and fees [viitattu 17.2.2016]. Saatavissa: <https://msdn.microsoft.com/library/windows/apps/jj863494.aspx>

Windows Dev Center. 2016b. Age ratings [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/mt148562.aspx>

Windows Dev Center. 2016c. App lifecycle [viitattu 2.3.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/hh464925.aspx>

Windows Dev Center. 2016d. App submissions [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/hh694062.aspx>

Windows Dev Center. 2016e. AppointmentStore class [viitattu 27.1.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.appointments.appointmentstore>

Windows Dev Center. 2016f. Choose Your Technology [viitattu 2.3.2016]. Saatavissa: [https://msdn.microsoft.com/en-us/library/windows/desktop/dn614993\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dn614993(v=vs.85).aspx)

Windows Dev Center. 2016g. Create your app by reserving a name [viitattu 11.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/jj657967.aspx>

Windows Dev Center. 2016h. Enter app properties [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/mt148533.aspx>

Windows Dev Center. 2016i. FindAppointmentsOptions class [viitattu 26.1.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/windows.applicationmodel.appointments.findappointmentsoptions.aspx>

Windows Dev Center. 2016j. Manage appointments [viitattu 27.1.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/xaml/mt269389.aspx>

Windows Dev Center. 2016k. Notes for certification [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/mt148541.aspx>

Windows Dev Center. 2016l. Opening a developer account [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/hh868184.aspx>

Windows Dev Center. 2016m. Resolve package upload errors [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/mt148544.aspx>

Windows Dev Center. 2016n. Set app pricing and availability [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/mt148548.aspx>

Windows Dev Center. 2016o. ShowReplaceAppointmentAsync [viitattu 27.1.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/dn624806>

Windows Dev Center. 2016p. The app certification process [viitattu 20.2.2016]. Saatavissa: <https://msdn.microsoft.com/en-us/library/windows/apps/mt148554.aspx>

Windows Phone How-to. 2016. Sync my calendars [viitattu 12.3.2016]. Saatavissa: <http://www.windowsphone.com/en-us/how-to/wp8/email-and-calendar/sync-my-calendars>

